

Cloud Computing Architecture

Technologies & Practice



顾炯炯 编著

云计算架构 技术与实践

清华大学出版社

Cloud Computing Architecture

Technologies & Practice



顾炯炯 编著

云计算架构 技术与实践

清华大学出版社

Cloud Computing Architecture

Technologies & Practice

顾炯炯 编著

云计算架构 技术与实践

清华大学出版社
北 京

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目（CIP）数据

云计算架构技术与实践/顾炯炯 编著. —北京：清华大学出版社，2014

ISBN 978-7-302-37820-4

I. ①云... II. ①顾... III. ①计算机网络—研究 IV. ①TP393

中国版本图书馆CIP数据核字（2014）第193817号

责任编辑：陈 莉 高 灿

封面设计：周晓亮

版式设计：方加青

责任校对：邱晓玉

责任印制：

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦A座

邮 编：100084

社总机： 010-62770175

邮 购： 010-62786544

投稿与读者服务： 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈： 010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载： <http://www.tup.com.cn>, 010-62794504

印刷者：

装订者：

经 销： 全国新华书店

开 本： 170mm×250mm（附光盘2张）

印 张： 20.25

字 数： 480千字

版 次： 2014年9月第1版

印 次： 2014年9月第1次印刷

印 数： 1~4000

定 价： 45.00元

产品编号：

编委会

第一作者： 顾炯炯

作者团队： 黄朝意、金波、李力、吴天议、王道辉、闵小勇、熊文辉、李浩、严天科、吴鸿钟、琚列丹、胡斐然、皮楚贤、陈普、李嘉、朱照生、周建军、刘红霞、肖江波、谢宁、孙万琪、潘少钦、胡善勇、王静、宁志强、张大震等

内容简介

云计算概念诞生至今已发展了约八年的时间。这八年来，相比云计算诞生初期，云计算技术条件、行业和市场环境均发生了巨大变化，广大读者对云计算的认知需求也从当初的粗浅概念阶段，发展到希望深度探索的阶段。

本书以云计算架构技术为核心，从讨论云计算发展为起点，围绕云计算架构涉及的核心技术与商业实践进行展开。论及的核心技术包括计算、存储、网络、数据、管理、接入、安全等，涵盖了云计算的最新趋势、原理、特性与实践。

本书适用于企业IT部门首席信息官、IT主管、IT技术工程师、技术类人员、IT技术公司员工、互联网公司员工、教育机构师生等。

作者简介

顾炯炯

华为公司云计算首席架构师，主导完成华为公司云计算操作系统（FusionCube）和融合基础设施（FusionCube）的技术规划与架构设计，支撑华为公司的ICT战略转型、云计算数据中心及电信云化解决方案，完成浙江移动、上海联通、上海健康云、新加坡StarHub、华为桌面云及数据仓库等数百个云计算项目的商用落地。曾任华为公司融合IMS解决方案首席架构师、华为公司移动软交换产品首席架构师，拥有已获授权并发布的个人专利30多项。

序言一

早在三千多年前，《易经》出现时，人类使用朴实的哲学思想揭示了一个变化的世界。变化无所不在，体现在政治、经济、社会、科学等方方面面。其中科学技术的发展变化，是这几个领域中最能被人们感同身受的。在科学技术领域，IT技术成为科技发展变化的急先锋。IT领域的“云计算”在2007年还是一个未知的概念，到2014年“云计算”不仅家喻户晓，而且在基于云计算的平台上创造了一个又一个发展速度的新记录：一款在云平台上的游戏，可以在几个月发展数千万用户；云平台支撑的电子商务，1秒钟可以完成数万笔交易。目前这种IT云计算领域的发展变化模式不再局限于令人瞠目的数字，而开始向传统行业乃至国家整体经济与社会领域进行快速渗透。我们相信，在不久的将来，全球的经济、社会与科学的面貌在云计算技术、服务与理念的推动下会焕然一新。这就是我们过去十几年常说的“信息技术革命”，在我们眼前正在发生、发展和不断演变着。

“云计算”这个名词虽然已经家喻户晓，而且业内基本认可美国国家标准与技术研究院（NIST）对云计算的概念定义，但人们对云计算的理解至今不同。究其根因，一方面是云计算的技术、服务模式和理念在不断演进和发展变化；另一方面，云计算的宣传推广主体，出于商业或理念的差异而对云计算的内涵进行了不同方向的强化和引申。人们对云计算的不同理解，势必引发概念定义上的争议，但一个不争的共识是：云计算已经落地生根，并快速地发展壮大，“像用电一样使用信息服务”的云计算理想虽然还未完全实现，但距离这个目标已经越来越近。我们每个人对云计算的发展阶段可能有不同的理解和划分，这很正常也很有益，因为多样化的观点碰撞是创新灵感的源泉。本书也将“云计算”的技术内涵进行了更大范围的发展和延伸。这让我联想到互联网行业正在发生的变化，早期的互联网公司仅仅开发与其直接相关的Web网站，但现如今，大型互联网公司所做的产品与业务早已远远地超出当初的范畴，不仅在看似简单的Web网站下面搭建了一个庞大的数据处理与分析平台，而且支撑其网站的服务器、存储、网络乃至数据中心。这些硬件产品均由互联网公司自己研发与设计，领先的互联网公司做的产品还远不止这些，Google推出了手机操作系统、眼镜，并致力于开发、完善无人驾驶汽车；亚马逊开发无人飞机用于物流投递；Facebook在虚拟现实领域进行了高额投入；国内的阿里巴巴、腾讯等公司在开发各种金融类创新产品。针对这些巨大的变化，在没有更好的名称之前，我们依然要称呼它

们是互联网公司，我们只能说当今互联网的含义比10年前的互联网更加宽泛。“云计算”也是如此。在IT领域，基本上每3~5年便会进行一次产品技术的更新换代，云计算经过多年发展，无论在技术深度还是在技术广度上均会有显著延展。通过阅读此书，可以明显感受到云计算技术这些年来发展和进步，以及云计算技术在企业IT和电信网络重构中的作用。

云计算不仅对企业和电信的发展有重大的推动作用，对教育科研领域也有深远的影响。“云计算”是由企业提出的概念，并一直由企业主导“云计算”技术的研发、推广与应用。云计算所支撑的大数据概念与技术的出现，企业主导，特别是互联网企业主导技术发展的趋势更加明显。企业之所以能够主导云计算与大数据技术的发展，原因可能来自三个方面：一是企业拥有雄厚的资金；二是云计算技术与企业应用及市场需求呈现出了紧耦合的发展；三是企业（特别是互联网企业）拥有足量的数据资源和计算资源。企业主导同时意味着在云计算与大数据领域传统的先有理论后有应用的模式基本不再适用，即大学科研机构负责理论（学术）创新，企业负责应用创新（工程实践）的分工模式。大学教育科研机构，特别是信息技术学科，需要调整自身角色来适应这一趋势的发展。云计算领域提供了非常丰富的开源环境，比如OpenStack、Hadoop、Xen、KVM等，互联网同时提供了数量庞大的信息技术文章，云计算厂商会提供免费或低价的试用软件或云计算环境，再加上不断出版更新的信息技术类书籍，这些资源构成了一个开放且实时更新的教学乃至实验环境，可以供广大学生学习和实践使用。未来大学对学生的培养工作可能集中在4个方面：方向性的指导，综合技能的认证，给学生提供专心学习交流的环境，为学生搭建创新的平台。大学的信息技术类学科的科研机构，需要与企业 and 市场更加紧密地结合，双方共享数据，共享研发测试环境，专利交叉共享，项目收益共享。大学科研机构可侧重承担企业偏远期的云计算与大数据技术的开发，企业侧重承担近期应用技术的开发，双方的界限会很模糊。在云计算时代，年龄、资历和职称都不再是科研创新的门槛，大学生可以广泛参与和承担云平台类产品应用开发，学生的自主创新成果（同时参照产品受市场欢迎的程度）将是个人能力认证的一个重要依据。

希望本书的所有读者，在了解云计算技术的同时，都能够积极地投身到云计算产业实践中来。只有更多的人认识到云计算的价值，才能挖掘出更多云计算的价值，云计算产业才会有源源不断的动力来蓬勃发展。相信各位读者中的很多人都将成为云计算产业的中坚力量。

李德毅

中国工程院院士，国际欧亚科学院院士

序言二

合作创新，云以致用

云计算从概念到大规模实践，短短数年间迅猛发展。它与诸多行业深度融合，带来了颠覆性的创新，凸显出巨大的应用价值和发展前景。读到华为公司云计算首席架构师顾炯炯新著《云计算架构技术与实践》，我倍感兴奋。著作概括了华为的云计算构想，表明华为云计算布局已走在行业的前列，也预示着英特尔和华为在云计算方面的合作将更加密切。

随着计算的延伸与扩展、移动互联网的发展，中国消费者拥有了越来越多的智能设备。为了让消费者在不同时间、不同场合获得实时信息服务，需要将各种内容和应用通过云端进行整合、匹配，推向不同的终端用户，实现从云到端高效顺畅的体验。从电信运营商到服务提供商，通过云架构部署各种移动设备和个性化服务，无疑是非常经济、便捷的途径。

同时，在经济发展和行业变革中，人们所遇到的挑战也日益复杂，需要综合的解决方案来应对。比如，在城镇化进程中，利用云计算结合物联网与大数据解决方案，可推动智能交通、平安城市、智慧医疗、环境监控等项目的建设，带来更好的城市运营管理和公共服务。将云计算与传统行业融合，将带来跨界创新，催生前所未有的商业模式、产业生态。云计算越来越从一种降低成本与提高效率的方式，演变成为向企业和消费者提供新服务的途径。

面对在线应用、服务和数据的高速增长，数据中心规模急剧膨胀，需要对数据中心IT基础设施的主要组成部分——服务器、存储和网络设备进行以软件定义为导向的创新，实现自动供给的IT资源池。英特尔于2013年开始践行“软件定义基础设施”战略，在开放平台上由用户自行定义他们的基础设施，这将使用户在云资源池内“调兵遣将”时更加轻松，进而也让他们的新应用和新服务更快地得到底层支持，实现迅速交付。

英特尔长期专注于计算的创新，提供从服务器到存储、网络全面的计算能力及软件优化支持，与产业伙伴共同应对各种云计算的发展难题，真正实现“云以致用”。为此，英特尔与产业界协同合作，基于开放架构推出了英特尔云构建计划，帮助开放数据中心联盟发掘和定义企业用户的

云计算需求，并提供丰富的云计算解决方案。

携手推动云计算创新，英特尔与华为的合作是一个很好的例证。华为不仅在通信行业处于领先地位，过去几年间其云计算业务也成倍增长。英特尔与华为的合作由来已久，双方都较早受到信息通信产业中最大的潮流——信息技术和通信技术融合的影响，合作领域不断拓展，从最初在通信领域，扩展到服务器、存储、网络领域，现在又把共同的目标放在了云计算和大数据的创新机会上。我们对双方的合作前景充满信心。

中国互联网用户众多，信息终端普及率很高，企业及消费者对于IT技术的了解和接受速度也非常快，再加上政府大力支持云计算产业，并实施宽带中国战略，这就使得中国在云计算部署规模、技术以及商业模式创新方面迅猛发展。我们期待着与国内合作伙伴深化和扩展合作领域，让更多源自中国的云计算创新成果去影响并推动全球信息技术和通信产业的发展进程！

杨叙

英特尔公司全球副总裁兼中国区总裁

前言

什么是云计算？美国国家标准与技术研究院（NIST）对此有这样一个权威和经典的定义：“所谓云计算，就是这样一种模式，该模式允许用户通过无所不在的、便捷的、按需获得的网络接入到一个可动态配置的共享计算资源池（其中包括了网络设备、服务器、存储、应用以及业务），并且以最小的管理代价或者业务提供者交互复杂度即可实现这些可配置计算资源的快速发放与发布。”

云计算的核心可以用五大基本特征、三种服务模式以及四类部署模式来概括，五大基本特征是按需获得的自助服务、广泛的网络接入、资源池化、快捷的弹性伸缩以及可计量的服务。三种服务模式为云基础设施即服务（IaaS）、云平台即服务（PaaS）以及云软件即服务（SaaS）。四类部署模式可以划分为专有云（私有云）、行业云、公有云以及混合云。

从各类云服务的创建、部署以及消费角度来描述云计算的实质，意味着云计算天然要求支持面向服务的能力。现代企业通常会将其IT基础设施、业务平台以及软件即服务的对外开放作为其整体端到端企业信息架构SOA解决方案中的重要一环来执行。当然，软件即服务（SaaS）作为一个流行多年的话题，最早出现是在云计算概念出现之前，已经不是什么新鲜概念了。

以亚马逊2006年3月13日发布的S3服务为起点，到“云计算”概念于2008年最早被Google提出，至今已有8年多的历史了，其核心理念已广为人们所传播和接受，也经历了方兴未艾的发展。当云计算还处于概念炒作的初级阶段时，云计算一度成为IT业界、媒体传播渠道乃至所有涉及IT信息化、政府宏观规划、关系国计民生的各大垂直行业关注的焦点，云计算也因此成为街头巷尾热议的“时髦”话题。与此同时，各种关于云计算的商业和解决方案也应运而生，各类理念和包装良莠不齐、不一而足，反而让大家对云计算到底能做什么，对其潜在的客户到底能够解决什么实际问题，能够带来什么样的实际价值感到迷惘，甚至使得大家对云计算的未来前景产生了怀疑。

关于云计算的社会价值与意义，我们常常用一句话表达云计算的目标诉求：“未来让人们像用水和用电那样使用云计算”。在这里，人们将云计

算视为一种“水和电”那样无处不在、人类社会日常生产和生活过程中必不可少的基础资源。这里我们用“电力”来形容云计算可能更为恰当，因为电力是用来驱动上一个工业文明时代最关键的生产资料。电灯、收音机、电视机、电冰箱、电风扇，乃至自动化生产线等无不需要依赖电力的驱动。相比电力，云计算则对应于当前的知识与信息时代进行任何信息分析与处理的生产资料，用于支撑ERP/CRM/Email/BI大数据乃至金融实时交易数据处理等所有维持企业业务正常运作所需的按需获取、按需分配的关键资源。

从技术架构演进的视角来看，有人将云计算视为自IT领域冯·诺依曼计算机架构诞生之后的第三次里程碑式的变革，是对传统计算架构与计算模式的颠覆与创新。也有人认为云计算无非是一种商业理念上的包装，所谓“新瓶装旧酒”，只是各个IT厂商用来“促销”自己产品的一种“营销活动”，并没有带来根本性的技术变革，也并没有给IT架构带来根本性的变化，那么真相究竟是怎样的呢？

回顾企业IT架构演进的整个历史，我们不难看到，冯·诺依曼架构的第一台计算机诞生以来的前30年，计算高度集中化、支持多用户多任务的大型机和小型机是企业IT的主流形态，构成IT系统的软硬件堆栈各层之间缺少统一的工业标准，呈现出内聚与耦合的特征，仅少数厂家拥有提供端到端高度复杂化的IT系统软硬件的能力。那个时代，IT系统造价高昂，往往是少数高端企业才能拥有的“奢侈品”。

于是，20世纪80年代，以x86服务器和PC系统的诞生为标志，企业IT系统迎来了第二次里程碑式的变革：从All in One、全封闭的软硬件栈走向了水平分层的网络、存储、服务器、操作系统、中间件、应用层等多层次水平分工的架构，各层之间接口标准化、规范化，极大地简化了每一层的技术复杂度，各层IT产业链获得了大繁荣与大发展，涌现出一批优秀的专业化厂家，聚焦于提供该领域内质量最佳的产品和解决方案，IT系统终于开始走入“寻常百姓家”。

然而，所谓“物极必反”，当这个架构分层发展到一定阶段，弊端逐步显现，由于企业IT层次太多，各层之间集成交付的难度越来越大，尤其是当今企业软件应用已从单一实例应用，迅速走向大规模分布式应用，一个关键业务的部署往往需要涉及服务器、网络、存储等各方面基础设施资源的协同配合，使得业务驱动的基础设施层服务器、存储、网络资源的集成管理配置和按需供给成为影响企业IT快速响应企业业务需求的关键制约因素。同时软硬件各层的开发虽然实现了解耦，但部署和运行态

仍然是软硬件耦合绑定的关系，因此跨服务器的资源出现忙闲不均时，依然无法有效利用IT资源。

随着企业信息化进程的不断推进，企业IT系统的使用者和维护者们逐渐发现，分层架构体系也存在着诸多弊端：

- 软硬件开发态解耦，但部署和运行态并未解耦；
- 生态链大繁荣的同时，多厂家硬件异构集成与管理的复杂度越来越高；
- 企业信息化的重心向软件转移，但计算、存储、网络硬件弹性供给能力及其相互协同的不足，越来越成为软件价值提升的制约性因素。

那么，是否存在一条IT架构演进路径，可以在代价最小化，即在不对现有软硬件堆栈做颠覆式改动的前提下，有效应对上述关键痛点与挑战呢？

答案是肯定的，这就是IT领域的第三次里程碑式演进变革，即从PC与服务器时代迈入云计算时代，通过虚拟化与云调度管理技术，将来自不同厂家的、多台烟囱式的、彼此孤立和割裂的计算、存储、网络设备在逻辑上整合成为一台“超大规模云计算机”，为上层的软件提供弹性的按需资源供给的能力，从而实现软硬件部署过程与运行态的解耦，屏蔽软硬件异构多厂家差异性与复杂度，并填补计算与存储之间的性能鸿沟。

其实大家也许已经注意到，我们谈到云计算驱动的第三次IT架构变革浪潮，其实早在云计算理念问世前的几年时间里，在众多互联网厂家中已被多次实践过，并且取得了巨大成功。那么普遍意义上的企业IT的云化重构又与互联网成功的实践之间存在着什么样的关联呢？

Google、Facebook的“云计算机”服务于其特定的商业模式和业务应用，比如搜索类、社交类应用，而企业IT云化架构所期望的“云计算机”，则面临着大量的、形形色色的面向传统IT基础设施架构开发的企业应用和电信应用，它们的应用场景需求既有相同点，也存在着巨大的差异化。

相同点在于：

- 计算、存储实现了大规模资源池化，实现了规模经济效益；
- 分布式架构与负载均衡能力，资源可按业务需求灵活扩展伸缩；
- 依赖分布式软件在系统整体层面而非单点硬件层面实现高可靠性及高性能保障。

不同点在于：

- 普适性——互联网平台一般仅为其特定业务模型定制，企业云平台则要求具备对异构多厂家应用的普遍适用性；
- 异构兼容性——企业云平台需要考虑异构厂家硬件的兼容性，需要实现对企业IT基础设施现有投资的最大化保护；
- 高性能——互联网业务虽然并发量和注册用户量庞大，但企业高端应用在时延和性能方面却有更高的要求；
- 自动化、虚拟化——互联网业务模式一般为自主开发、自运主营（DevOps、Development和Operations的组合），因此对管理自动化要求不迫切，企业应用则由于应用颗粒度不一，基础设施采购自第三方，因此管理自动化和虚拟化基本为必选能力。

“天下大势，合久必分，分久必合”，云计算时代IT基础设施演进的下一个十年，是从分离重新走向融合的十年：

- 通过云操作系统，将数据中心多厂家异构的计算、存储、网络资源进行水平融合，对外提供开放与标准化的IT服务接口，实现面向利用IT基础设施的“融合”；
- 通过融合架构一体机，将单厂家计算、存储与网络资源进行垂直融合，提供模块化、一站式、高性能、性价比最优、面向新建IT基础设施的交付模式。

无论IT架构如何螺旋式演进，客户价值和驱动力都体现在：

- 更低的TCO ^①

- 更高的业务部署与生命周期管理效率；
- 更优的业务性能与用户体验。

IT基础设施架构从分离重新走向融合，并非简单的历史重复，而是在继承现有成果的基础上创新突破。无论是水平融合，还是垂直融合，在核心技术支撑方面并未将现有已形成产业规模的x86 CPU及其服务器计算架构推倒重来，而是在最大限度地重用这些成熟产业组件的前提下，借助虚拟化及分布式云计算调度管理软件的作用，将多厂家异构或者单厂家同构的计算、存储、网络整合为规模可大可小的“云计算机”，从而有效地解决传统IT架构所面临的挑战——业务上线周期长，TCO居高不下，企业关键应用性能低下。

顾炯炯

① TCO，总拥有成本，即从产品采购到后期使用、维护的总成本。

目 录

[编委会](#)

[内容简介](#)

[作者简介](#)

[序言一](#)

[序言二](#)

[前 言](#)

[第1章 云计算理念的发展](#)

[1.1 云计算的基础概念与架构](#)

[1.2 云计算的发展趋势](#)

[第2章 云计算的架构内涵与关键技术](#)

[2.1 云计算的总体架构](#)

[2.1.1 云计算核心架构上下文](#)

[2.1.2 云计算平台架构](#)

[2.2 云计算架构的关键技术](#)

[2.2.1 超大规模资源调度算法](#)

[2.2.2 异构集成技术](#)

[2.2.3 应用无关的可靠性保障技术](#)

[2.2.4 单VM及多VM的弹性伸缩技术](#)

[2.2.5 计算近端I/O性能加速技术](#)

[2.2.6 网络虚拟化技术](#)

[2.2.7 应用管理自动化技术](#)

[2.3 云计算核心架构的竞争力衡量维度](#)

[2.3.1 低TCO](#)

[2.3.2 弹性伸缩](#)

[2.3.3 高性能](#)

[2.3.4 领先的用户体验](#)

[2.3.5 高安全](#)

[2.3.6 高可靠](#)

[2.4 云计算解决方案的典型架构组合及落地应用场景](#)

[2.4.1 桌面云](#)

[2.4.2 存储云](#)

[2.4.3 IDC托管云](#)

[2.4.4 企业私有云](#)

[2.4.5 大数据分析云](#)

[2.4.6 数据库云](#)

[2.4.7 媒体云](#)

[2.4.8 电信NFV云](#)

[第3章 云计算相关的开源软件](#)

[3.1 云计算领域开源软件概览](#)

[3.2 Cloud OS开源软件：CloudStack](#)

[3.2.1 CloudStack的总体架构](#)

[3.2.2 CloudStack的资源管理](#)

[3.2.3 CloudStack的虚拟机管理](#)

[3.2.4 CloudStack的块存储管理](#)

[3.2.5 CloudStack的虚拟网络](#)

[3.3 Cloud OS开源软件：OpenStack](#)

[3.3.1 OpenStack的总体架构](#)

[3.3.2 OpenStack的计算服务：Nova](#)

[3.3.3 OpenStack的块存储服务：Cinder](#)

[3.3.4 OpenStack的网络服务：Neutron](#)

[3.3.5 OpenStack的镜像服务：Glance](#)

[3.3.6 OpenStack的身份服务：KeyStone](#)

[3.4 开源和社区发展](#)

[3.4.1 Hypervisor社区发展](#)

[3.4.2 Cloud OS社区发展](#)

[3.5 开源还是闭源](#)

[第4章 面向电信及企业关键应用的计算虚拟化](#)

[4.1 计算虚拟化核心引擎：Hypervisor介绍](#)

[4.1.1 业界典型计算虚拟化架构说明](#)

[4.1.2 满足电信和企业关键应用的计算虚拟化技术](#)

[4.2 跨服务器的计算资源调度算法](#)

[4.2.1 高性能、低时延的虚拟机热迁移机制](#)

[4.2.2 计算资源池的动态资源调度管理和动态能耗管理](#)

[4.3 计算高可靠性保障](#)

[4.3.1 基于冷备机制的虚拟机HA保护](#)

[4.3.2 基于热备机制的虚拟机运行业务镜像冗余方案](#)

[4.3.3 无状态计算及物理机可靠性保障](#)

[第5章 面向网络自动化、多租户的网络虚拟化](#)

[5.1 网络虚拟化的驱动力与关键需求](#)

[5.2 SDN架构](#)

[5.2.1 IETF定义的SDN架构介绍](#)

[5.2.2 ONF OpenFlow网络架构](#)

[5.2.3 OpenFlow协议介绍](#)

[5.2.4 OF-Config](#)

[5.2.5 ONF及OpenDayLight标准联盟](#)

[5.3 网络虚拟化关键技术：大二层实现](#)

[5.3.1 CT流派：以交换机为中心的大二层技术](#)

[5.3.2 IT流派：以服务器叠加网为中心的Overlay技术](#)

[5.4 网络虚拟化关键技术：多租户网络实现](#)

[5.5 网络虚拟化端到端解决方案](#)

[5.6 网络云化还有多远](#)

[第6章 面向企业关键应用性能提升和存储管理简化的存储虚拟化](#)

[6.1 云计算的存储虚拟化概述](#)

[6.2 灵活的软件定义存储](#)

[6.3 传统存储SAN/NAS的虚拟化](#)

[6.4 分布式存储池化和加速](#)

[6.4.1 分布式存储概述](#)

[6.4.2 分布式存储系统的架构](#)

[6.4.3 分布式存储关键技术：性能提升技术](#)

[6.4.4 分布式存储关键技术：简化管理技术](#)

[6.4.5 分布式存储关键技术：安全可靠性能增强技术](#)

[第7章 云接入的关键技术架构与应用](#)

[7.1 云接入的概述](#)

[7.1.1 什么是云接入](#)

[7.1.2 云接入的作用和意义](#)

[7.1.3 云接入的挑战和需求](#)

[7.2 云接入的架构](#)

[7.3 云接入的典型应用](#)

[7.3.1 桌面云的概念和价值](#)

[7.3.2 桌面云的逻辑架构](#)

[7.3.3 桌面云典型应用场景](#)

[7.3.4 移动办公的概念和价值](#)

[7.3.5 移动办公的逻辑架构](#)

[7.3.6 移动办公解决方案的特点](#)

[7.4 云接入的关键技术](#)

[7.4.1 桌面云协议简介](#)

[7.4.2 桌面云协议关键技术：高效远程显示](#)

[7.4.3 桌面云协议关键技术：低资源消耗的多媒体视频](#)

[7.4.4 桌面云协议关键技术：低时延音频](#)

[7.4.5 桌面云协议关键技术：兼容多种外设](#)

[7.4.6 桌面云协议总结与其他实现](#)

[7.5 云接入的发展趋势](#)

[7.5.1 云接入的未来发展](#)

[7.5.2 VDI](#)

[7.5.3 DaaS](#)

[7.5.4 移动办公](#)

[第8章 云管理与自动化的关键技术架构与应用](#)

[8.1 业务应用驱动的拉通计算、存储、网络自动化](#)

[8.1.1 自动化部署](#)

[8.1.2 动态调度](#)

[8.1.3 网络自动化](#)

[8.2 物理和虚拟化资源的统一管控](#)

[8.2.1 物理资源管理](#)

[8.2.2 虚拟化资源管理](#)

[8.2.3 资源集群管理](#)

[8.2.4 虚拟机资源管理](#)

[8.3 基于网络的硬件即插即用的自动化机制](#)

[8.3.1 设备自动发现和部署](#)

[8.3.2 服务器自动化](#)

[8.4 异构硬件的统一接入管理](#)

[8.5 服务目录和应用管理](#)

[8.5.1 应用发布流程介绍](#)

[8.5.2 应用管理原理](#)

[8.6 面向云管理的ITSM](#)

[8.7 云平台第三方App资源使用计量](#)

[8.8 云管理的应用案例](#)

[8.8.1 M运营商私有云建设](#)

[8.8.2 T运营商分布式数据中心](#)

[8.8.3 新加坡S运营商中小企业IT应用托管](#)

[第9章 云安全架构与应用实践](#)

[9.1 端到端云安全架构](#)

[9.1.1 云计算中的主要安全威胁](#)

[9.1.2 端对端的安全架构](#)

[9.2 可信计算TPM/vTPM](#)

[9.2.1 TPM功能1：主机启动/静态度量](#)

[9.2.2 TPM功能2：虚拟机的静态度量](#)

[9.2.3 TPM功能3：主机动态度量](#)

[9.2.4 TPM功能4：VM动态度量](#)

[9.2.5 TPM功能5：远程证明](#)

[9.3 虚拟机的安全隔离](#)

[9.3.1 vCPU调度隔离安全](#)

[9.3.2 内存隔离](#)

[9.3.3 内部网络隔离](#)

[9.3.4 磁盘I/O隔离](#)

[9.3.5 用户数据隔离](#)

[9.4 虚拟化环境中的网络安全](#)

[9.4.1 虚拟交换机及防ARP攻击](#)

[9.4.2 IP/MAC防欺骗功能设计](#)

[9.4.3 VLAN](#)

[9.5 云数据安全](#)

[9.5.1 云存储加密与用户数据安全](#)

[9.5.2 用户数据安全有效保护](#)

[9.6 公有云、私有云的安全组](#)

[9.7 云安全管理](#)

[9.7.1 日志管理](#)

[9.7.2 账户和密码安全](#)

[9.7.3 分权分域管理](#)

[9.8 云安全应用实施案例](#)

[9.9 云计算安全的其他考虑](#)

[第10章 大数据平台核心技术与架构](#)

[10.1 大数据特点与支撑技术](#)

[10.1.1 数据采集技术](#)

[10.1.2 数据预处理技术](#)

[10.1.3 数据存储及管理技术](#)

[10.1.4 数据分析及挖掘技术](#)

[10.1.5 数据展现与应用技术](#)

[10.2 企业级Hadoop](#)

[10.2.1 Apache Hadoop起源](#)

[10.2.2 企业级Hadoop总体框架](#)

[10.2.3 HDFS](#)

[10.2.4 MapReduce](#)

[10.2.5 ZooKeeper](#)

[10.2.6 HBase](#)

[10.2.7 Hive](#)

[10.3 流处理技术](#)

[10.3.1 流处理的应用场景](#)

[10.3.2 流处理技术的关键概念](#)

[10.3.3 流处理技术的辨析](#)

[10.3.4 流处理技术的最新发展](#)

[10.3.5 分布式事件的流处理技术](#)

[10.4 大数据在金融领域的探索与实践](#)

[10.4.1 银行业现状和大数据的潜在机会](#)

[10.4.2 大数据时代的银行业发展](#)

[10.4.3 大数据在银行业的发展趋势](#)

[10.4.4 大数据在金融行业的实践](#)

[10.5 未来大数据应用畅想](#)

[10.5.1 身边的大数据](#)

[10.5.2 大数据将重构很多行业的商业思维和商业模式](#)

[第11章 企业私有云和公有云对IAAS层的诉求](#)

[11.1 企业私有云和公有云对IAAS层的诉求](#)

[11.2 一体机的市场和技术](#)

[11.2.1 一体机市场](#)

[11.2.2 一体机技术](#)

[11.2.3 一体机产品介绍](#)

[11.3 一体机市场、技术趋势](#)

[11.3.1 一体机市场趋势](#)

[11.3.2 一体机技术趋势](#)

[结 语](#)

[缩略语](#)

[后 记](#)

[附录CD](#)

第1章 云计算理念的发展

1.1 云计算的基础概念与架构

过去20年内，全球无线与宽带技术及其商业化应用部署获得了长足的发展与进步，机器与机器之间、机器与人之间的网络连接不再是稀缺资源和瓶颈，为IT计算架构从本地计算模式及客户端、服务器端并重的传统模式，向以“广泛的网络接入”、“计算、存储的集中资源池化”、“快捷的弹性伸缩”、“按需自助及可计量的服务”为典型特征的云计算模式的演进铺平了道路，并掀起了正在席卷全球的第三次IT变革浪潮。

在传统IT体系架构下，当前企业基础设施建设与运维所面临的核心痛点问题可以概括为如下几点。

平均资源利用率及能耗效率低下

针对基础设施平台建设、扩容与更新换代，当前企业普遍采用的模式是服务器、网络交换与安全以及存储设备的水平分层采购。各个IT基础设施单部件的选型、数量以及不同部件的组网连接方案均取决于企业IT收集的各业务部门对于IT核心业务处理量需求的预测和规划。同时，所有企业IT应用软件、数据库以及中间件软件均采用独占计算、存储和网络资源的烟囱式部署。软件应用与硬件唯一捆绑，不同应用之间无法动态、高效共享相同的计算与存储资源。加之按照摩尔定律不断翻番增长的CPU计算能力已大大超出应用软件对计算资源利用率的同步能力，导致企业IT的平均资源利用率始终处在低于20%的水平。

新业务上线测试周期长，效率低下

企业任何一项新业务上线，从最基础的硬件平台开始，向上逐层延伸至操作系统、中间件、数据库以及CRM/ERP/HRM/PDM/Email/UC等各类业务关键软件堆栈，这一过程需要投入IT专业化团队，进行软件安装、调试、功能与性能验证测试、网络配置及修改调整，然后经过若干轮测试、故障及性能稳定性测试定位及重配置和调整之后，才能最终达到期望正式上线运行的成熟度水准。这个过程一般需要长达2至3个月的时间。

资源储备及弹性伸缩能力不足，不具备应对企业**IT**突发业务高峰处理的能力

针对特定垂直行业，短时间内突发性的**高流量、高密度业务需求**（比如节假日期间对视频网站的突发业务流程冲击），企业内部物理基础设施资源往往无法满足短时间内迅速获取所需资源的需求，以及处置业务高峰过后的资源闲置问题。

企业核心信息资产存在通过个人**PC**电脑/便携设备外泄的安全风险，无法在个人智能终端（平板电脑、智能手机）方便地访问企业防火墙内的工作流及文档

部分企业核心信息资产通过员工个人**PC**电脑或便携设备外泄给竞争对手，给企业竞争力和商业利益带来负面影响。过分严格的信息安全管控措施又导致了工作效率的下降，企业管理层及员工无法便捷地通过无所不在的网络访问企业防火墙内部的信息资产。

中小型企业希望通过宽带网络管道，从电信运营商或其他主机托管运营商的托管应用数据中心“按需获取”其所需的企业**IT**应用能力，从而实现日常运作中**IT**成本开销最小化

数量众多的中小企业，缺少**IT**领域专业经验，甚至没有财力和精力建设和维持自己专属的**IT**部门以及**IT**基础设施平台，普遍希望可以直接从托管运营商那里获取支撑其日常业务运作所需的**SaaS**服务。

针对解决上述企业**IT**系统建设和维护过程中遇到的普遍问题，迫切呼唤业界**IT**软硬件解决方案提供商借助云计算技术，打造**TCO**、性价比与效率最优的“**IT**基础设施私有云及公有云”，具体包括：

- 面向大型企业和行业领域提供全自动化管理、一站式交付、支持与企业**ITIL**无缝集成融合、**TCO**最优化的端到端解决方案，实现企业传统**IT**基础设施的改造、扩容和新建；

- 面向中小型企业（**SMB**），提供支持多租户安全隔离与动态发放、超大规模资源池调度管理、可最大限度地发挥规模经济效益的

公有云托管解决方案。

无论上述哪一类形态，企业云计算IT基础设施平台均可定位于基础设施、中间层云平台服务、云计算业务发放与维护管理。针对云平台服务层之上的多样化的内部IT软件及外部增值业务软件，企业（含运营商）可奉行“深淘滩、低作堰”的原则，广结各方ISV合作联盟，建设依托于云计算平台、繁荣的企业私有云及公有云生态系统。

通过上述私有云/公有云生态系统的建设，使得企业及运营商客户可以真正将“IT基础设施平台”与“核心业务流程”及“对外服务”解耦，大幅精简企业、运营商的内部IT及对外业务的基础设施层建设部署、运营维护及生命周期管理成本，从而更好地聚焦“核心业务流程”及“对外服务”的开发与定制，帮助企业和运营商在新形势下获得可持续发展。用一句话高度概括企业云计算IT基础设施平台的核心价值就是：“精简IT，敏捷商道”。

1.2 云计算的发展趋势

由于技术方面的限制，前面几年云计算的主要应用仍然仅仅局限在互联网领域以及规模有限的公有云的建设方面，而在私有云的建设方面，往往仅仅将建设聚焦在企业内部的生产云和测试云，以及外围办公桌面系统上。

随着云计算技术日新月异的发展，以及信息产业界不懈的“化云为雨”的努力，云计算迈入了一个新的阶段，云计算在企业信息化以及电信网络转型变革中进行了全面渗透与应用落地。

总体来说，新阶段云计算区别于以往的关键差别体现在以下几个方面。

差别1：从IT非关键应用走向电信网络应用和企业关键应用

站在云计算面向企业IT及电信网络的使用范围的视角来看，云计算发展初期，虚拟化技术主要局限于非关键应用，比如办公桌面云、开发测试云等。该阶段的应用往往对底层虚拟化带来的性能开销并不敏感，人们更加关注于资源池规模化集中之后资源利用效率的提升以及业务部署效率的提升。然而，随着云计算的持续深入普及，企业IT云化的范围已从

周边软件应用，逐步走向更加关键的企业应用，甚至企业的核心生产IT系统。由此，如何确保云平台可以更为高效、更为可靠地支撑好时延敏感的企业关键应用，就变得至关重要。

对于企业IT基础设施的核心资产而言，除去实实在在的计算、存储、网络资源等有形物理资产之外，最有价值的莫过于企业数据这些无形资产。在云计算的计算虚拟化技术发展初级阶段，Guest OS与Host OS之间的前后端I/O队列在I/O吞吐上的开销较大，而传统的结构化数据由于对I/O性能吞吐和时延要求很高，这两个原因导致很多事务关键型结构化数据在云化的初级阶段并未被纳入虚拟化改造的范畴，从而使得相关结构化数据的基础设施仍处于虚拟化乃至云计算资源池的管理范围之外。然而随着虚拟化XEN/KVM引擎在I/O性能上的不断优化提升（如采用SR-IOV直通、多队列优化技术），使得处于企业核心应用的ERP等关系型关键数据库迁移到虚拟化平台上实现部署和运行已不是问题。

与此同时，云计算在最近2~3年内，已从概念发源地的互联网IT领域渗透到电信运营商网络领域。互联网商业和技术模式的成功，启发电信运营商们通过引入云计算实现对现有电信网络和网元的重构来打破传统意义上电信厂家所采用的电信软件与电信硬件绑定的销售模式，同样享受到云计算为IT领域带来的红利，如：硬件TCO的降低、绿色节能、业务创新和部署效率的提升、对多国多子网的电信功能的快速软件定制化以及更强的对外能力开放。

差别2：从计算虚拟化走向存储虚拟化和网络虚拟化

从支撑云计算按需、弹性分配资源，与硬件解耦的虚拟化技术的角度来看，云计算早期阶段主要聚焦在计算虚拟化领域。事实上，众所周知的计算虚拟化技术早在IBM 370时代就已经在其大型机操作系统上诞生了。技术原理是通过在OS与裸机硬件之间插入虚拟化层，来在裸机硬件指令系统之上仿真模拟出多个370大型机的“运行环境”，使得上层“误认为”自己运行在一个独占系统之上，实际上是由计算虚拟化引擎在多个虚拟机之间进行CPU分时调度，同时对内存、I/O、网络等访问进行访问屏蔽。当x86平台演进成为在IT领域硬件平台的主流之后，VMware ESX、XEN、KVM等依托于单机OS的计算虚拟化技术才将IBM 370的虚拟化机制在x86服务器的硬件体系架构下实现并进行商品化，并且在单机/单服务器虚拟化的基础上，引入了具备虚拟机动态迁移和HA调度能力的中小集群管理软件（如vCenter/vSphere、XEN Center、Fusion

Sphere等），从而形成了当前的计算虚拟化主体。

随着数据和信息越来越成为企业IT中最为核心的资产，作为数据信息持久化载体的存储已经逐步从服务器计算中剥离出来，成为一个庞大的独立产业，与必不可少的CPU计算能力一样，在数据中心发挥着至关重要的作用。当企业对存储的需求发生变化时该如何快速满足新的需求以及如何利用好已经存在的多厂家的存储，这些问题都需要存储虚拟化技术来解决。

与此同时，现代企业数据中心的IT硬件的主体已经不再是封闭的、主从式架构的大型机、小型机一统天下的时代。客户端与服务器之间南北方向通信、服务器与服务器之间东西方向协作通信以及从企业内部网络访问远程网络和公众网络的通信均已走入了基于对等、开放为主要特征的以太互联和广域网互联时代。因此，网络也成为计算、存储之后，数据中心IT基础设施中不可或缺的“三要素”之一。

就企业数据中心端到端基础设施解决方案而言，服务器计算虚拟化已经远远不能满足用户在企业数据中心内对按需分配资源、弹性分配资源、与硬件解耦的分配资源的能力需求，由此存储虚拟化和网络虚拟化技术应运而生。

除去云管理和调度所完成的管理控制面的API与信息模型归一化处理之外，虚拟化的重要特征是通过在指令访问的数据面上，对所有原始访问命令字进行截获，并实时执行“欺骗”式仿真动作，使得被访问的资源呈现出与其真正的物理资源不同的（软件无需关注硬件）、“按需获取”的颗粒度。对于普通x86服务器来说，CPU和内存资源虚拟化后再将其（以虚拟机CPU/内存规格）按需供给资源消费者（上层业务用户）。由于计算能力的快速发展，以及软件通过负载均衡机制进行水平扩展的能力提升，计算虚拟化中仅存在资源池的“大分小”的问题。然而对于存储来说，由于最基本的硬盘（SATA/SAS）容量有限，而客户、租户对数据容量的需求越来越大，因此必须考虑对数据中心内跨越多个松耦合的分布式服务器单元内的存储资源（服务器内的存储资源、外置SAN/NAS在内的存储资源）进行“小聚大”的整合，组成存储资源池。这个存储资源池，可能是某一厂家提供的存储软硬件组成的同构资源池，也可能是被存储虚拟化层整合成为跨多厂家异构存储的统一资源池。各种存储资源池均能以统一的块存储、对象存储或者文件的数据面格式进行访问。

对于数据中心网络来说，网络的需求并不是凭空而来的，而是来源于业务应用，与作为网络端节点的计算和存储资源有着无法切断的内在关联性。然而，传统的网络交换功能都是在物理交换机和路由器设备上完成的，网络功能对上层业务应用而言仅仅体现为一个一个被通信链路连接起来的孤立的“盒子”，无法动态感知来自上层业务的网络功能需求，完全需要人工配置的方式来实现对业务层网络组网与安全隔离策略的需要。在多租户虚拟化的环境下，不同租户对于边缘的路由及网关设备的配置管理需求存在极大的差异化，而物理路由器和防火墙自身的多实例能力也无法满足云环境下租户数量的要求，采用与租户数量等量的路由器与防火墙物理设备，成本上又无法被多数客户所接受。于是人们思考是否可能将网络自身的功能从专用封闭平台迁移到服务器通用x86平台上来。这样至少网络端节点的实例可以由云操作系统来直接自动化地创建和销毁，并通过一次性建立起来的物理网络连接矩阵，进行任意两个网络端节点之间的虚拟通信链路建立，以及必要的安全隔离保障，从而里程碑式地实现了业务驱动的网络自动化管理配置的能力，大幅度降低数据中心网络管理的复杂度。从资源利用率的视角来看，任意两个虚拟网络节点之间的流量带宽，都需要通过物理网络来交换和承载，因此只要不超过物理网络的资源配额上限（默认建议物理网络按照无阻塞的CLOS模式来设计实施），只要虚拟节点被释放，其所对应的网络带宽占用也将被同步释放，因此也就相当于实现对物理网络资源的最大限度的“网络资源动态共享”。换句话说，网络虚拟化让多个盒子式的网络实体第一次以一个统一整合的“网络资源池”的形态出现在业务应用层面前，同时与计算和存储资源之间也有统一协同机制。

差别3：资源池从小规模的资源虚拟化整合走向更大规模的资源池构建，应用范围从企业内部走向多租户的基础设施服务乃至端到端IT服务

站在云计算提供像用水用电一样方便的服务能力的技术实现角度来看，云计算发展早期，虚拟化技术（如VMware ESX，微软Hyper-V，基于Linux的XEN、KVM）被普遍采用，用来实现以服务器为中心的虚拟化资源整合。在这个阶段，企业数据中心中的服务器只是部分孤岛式的虚拟化以及资源池整合，还没有明确的多租户以及服务自动化的理念。在该阶段，服务器资源池整合的服务对象是数据中心的基础设施硬件以及应用软件的管理人员。在实施虚拟化之前，物理的服务器及存储、网络硬件是数据中心管理人员的管理对象，在实施虚拟化之后，管理对象从物理机转变为虚拟机及其对应的存储卷、软件虚拟交换机，甚至软件防

防火墙功能。目标是实现多应用实例和操作系统软件在硬件上最大限度地共享服务器硬件，通过多应用负载的削峰错谷达到资源利用率提升的目的，同时为应用软件进一步提供额外的HA/FT（High Availability/Fault Tolerance，高可用性/容错）可靠性保护，以及通过轻载合并、重载分离的动态调度，对空载服务器进行下电控制，实现PUE功耗效率的优化提升。

然而，这些虚拟化资源池的构建，仅仅是从数据中心管理员视角实现了资源利用率和能效比的提升，与真正的多租户云服务模式仍然相差甚远。因为在云计算进一步走向普及深入的新阶段，通过虚拟化整合之后的资源池的服务对象，不能再仅仅局限于数据中心管理员本身，而需要扩展到每个云租户。因此云平台必须在基础设施资源运维监控管理Portal的基础上，进一步面向每个内部或者外部的云租户提供按需定制的基础设施资源，订购与日常维护管理的Portal或者API界面，并将虚拟化或者物理的基础设施资源的增、删、改、查等权限按照分权分域的原则赋予每个云租户，每个云租户仅被授权访问其自己申请创建的计算、存储以及与相应资源附着绑定的OS和应用软件资源，最终使得这些云租户可以在无需购买任何硬件IT设备的前提下，实现按需快速资源获取，以及高度自动化部署的IT业务敏捷能力的支撑，从而将云资源池的规模经济效益和弹性按需的快速资源服务的价值充分发掘出来。

差别4：数据规模从小规模走向海量，数据形态从传统结构化走向非结构化和半结构化

站在云计算系统需要提供的处理能力角度看，随着智能终端的普及、社区网络的火热、物联网的逐步兴起，IT网络中的数据形态已经由传统的结构化、小规模数据，迅速发展成为有大量文本、大量图片、大量视频的非结构化和半结构化数据，数据量也呈几何级增长。

对非结构化、半结构化大数据的处理而产生的数据计算和存储量的规模需求，已远远超出传统的Scale-up硬件系统可以处理的，因此要求必须充分利用云计算提供的Scale-out架构特征，按需获得大规模资源池来应对大数据的高效高容量分析处理的需求。

企业内日常事务交易过程中积累的大数据或者从关联客户社交网络以及网站服务中抓取的大数据，其加工处理往往并不需要实时处理，也不需要系统处于持续化的工作态，因此共享的海量存储平台，以及批量并行

计算资源的动态申请与释放能力，将成为未来企业以最高效的方式支撑大数据资源需求的解决方案备选项。

差别5：企业应用接入模式从传统接入走向BYOD接入

从云计算系统给终端用户提供的接入能力角度看，云计算架构下，随同企业用户近端计算、存储资源向远端的数据中心的迁移和集中化部署，带来了企业用户如何通过企业内部局域网及外部固定、移动宽带广域网等多种不同途径，借助固定、移动等多种不同瘦终端或智能终端形态接入云端企业应用的问题。面对局域网及广域网连接在通信包转发与传输时延不稳定、丢包以及端到端QoS质量保障机制缺失等实际挑战，如何确保远程云接入的性能体验达到与本地计算相同的水平，成为企业云计算IT基础设施平台面临的又一大挑战。

为应对云接入管道上不同业务类型对业务体验的不同诉求，解决业界通用远程桌面协议（如RDP）在满足本地计算体验能力方面存在的缺陷与不足，需重点关注ALL IP多媒体音视频端到端QoS/QoE优化，尤其是在远程桌面接入协议中对不同业务类别进行动态识别并区别处理，使其满足如下场景需求。

➤ 普通办公业务响应时延小于100ms，带宽占用小于150Kbps：通过在服务器端截获GDI/DX/OpenGL绘图指令，结合对网络流量的实时监控和分析，从而选择最佳的传输方式和压缩算法，将服务端绘图指令重定向到瘦客户端或软终端重放，从而实现时延与带宽占用的最小化。

➤ 针对虚拟桌面环境下VoIP质量普遍不佳的情况，默认的桌面协议TCP连接不适合作为VoIP承载协议的特点：采用RTP/UDP代替TCP，并选择G.729/AMR等成熟的VoIP Codec；在瘦客户端可以支持VoIP/UC客户端的情况下，尽量引入VoIP虚拟机旁路方案，从而减少不必要的额外编解码处理带来的时延及话音质量上的开销；上述优化措施使得虚拟桌面环境下的话音业务MOS平均评估值从3.3提升到4.0。

➤ 针对远程云接入的高清（1080p/720p）视频播放场景：在云端桌面的多虚拟机并发且支持媒体流重定向的场景下，针对普通瘦终端高清视频解码处理能力不足的问题，桌面接入协议客户端软件应

具备通过专用API调用瘦终端芯片（ARM/ATOM）多媒体硬解码处理能力；部分应用如Flash以及直接读写显卡硬件的视频软件，必须依赖GPU或硬件DSP的并发编解码能力，基于通用CPU的软件编解码将导致画面停滞，体验无法接受，可选择由硬件GPU虚拟化或DSP加速卡来有效提升云端高清视频应用的访问体验，达到与本地视频播放相同的清晰与流畅度。桌面协议还能够智能识别并区分画面变化热度，仅对变化度高且绘图指令重定向无法覆盖部分才启动带宽消耗较高的显存数据压缩重定向。

➤ 针对工程机械制图、硬件PCB制图、3D游戏、VR仿真，甚至Win7 Aero透明效果等云端图形计算密集型类应用：需要虚拟化GPU资源进行硬件辅助的渲染与压缩加速处理。

另一方面，正当全球消费者IT步入方兴未艾的Post-PC时代大门之时，iOS及Android智能终端正在悄悄取代企业用户办公位上的PC甚至便携电脑，企业用户希望通过智能终端不仅可以方便地访问传统Windows桌面应用，还可以从统一的“桌面工作空间”访问公司内部的Web SaaS应用、第三方的外部SaaS应用以及其他Linux桌面系统里的应用，而且希望一套企业的云端应用可以不必针对每类智能终端OS平台开发多套程序，就能够提供覆盖所有智能终端形态的统一业务体验，基于此考虑，企业云计算IT基础设施需要在面向传统Windows桌面应用云接入的自研桌面协议基础上，引入基于HTML5协议、支持跨多种桌面OS系统、支持统一认证及应用聚合、支持应用零安装升级维护以及异构智能终端多屏接入统一体验的云接入解决方案——Web Desktop。

差别6：云平台从闭源、封闭走向开源、开放

从云计算平台的接口兼容能力角度看，云计算早期阶段，闭源VMware vSphere/ vCenter、微软SystemCenter/Hyper-V云平台软件由于其虚拟化成熟度遥遥领先于开源云平台软件的成熟度，因此导致闭源的私有云平台成为业界主流的选择。然而，随着XEN/ KVM虚拟化开源，以及OpenStack、CloudStack、Eucalyptus等云操作系统OS开源软件系统的崛起和快速进步，开源力量迅速发展壮大起来，迎头赶上并逐步成长为可以左右行业发展格局的重要决定性力量。仅以OpenStack为例，目前IBM、HP、Suse、Redhat、Ubuntu等领先的软硬件公司都已成为OpenStack白金会员，从2010年诞生第一个版本开始，平均每半年发布一个新版本，所有会员均积极投身到开源贡献中来，到目前为止已推出

8个版本（A/B/C/D/E/F/G/H/I），功能不断完善。到2014年上半年，OpenStack的成熟度与vCloud/ vSphere 5.0版本的水平相当，满足基本规模商用和部署要求。从目前的发展态势来看，OpenStack开源大有成为云计算领域的Linux开源之势。回想2001年前后当Linux OS仍相当弱小、UNIX操作系统大行其道且占据企业IT系统主要生产平台的阶段时，多数人不会想象到仅10年时间，Linux已取代UNIX，成为主导企业IT服务端的默认OS选择，开源Linux正在配合Intel x86来取代小型机、大型机。

本书下面的内容将重点围绕云计算出现的这些新变化来讲述云计算的架构技术。

第2章 云计算的架构内涵与关键技术

2.1 云计算的总体架构

从上述分析不难看出，云计算推动了IT领域自20世纪50年代以来的第三次变革浪潮，对各行各业数据中心基础设施的架构演进及上层应用与中间件层软件的运营管理模式产生深远影响。在云计算发展早期，

Google、Amazon、Facebook等互联网巨头们在其超大规模Web搜索、电子商务及社交等创新应用的牵引下，树立了业界云计算平台基础架构的标杆与典范，但那个时期，多数行业与企业IT的数据中心仍然是基于传统的以硬件资源为中心的架构，即便是已进行了部分云化的探索，也多为新建的孤岛式虚拟化资源池（如基于VMware的服务器资源整合），或者仅仅对原有软件系统的服务器进行虚拟化整合改造。随着云计算技术与市场的发展，近两年云计算技术与架构才开始在各行各业信息化建设中和数据中心的演进变革中逐步得到了真正广泛和全面的落地部署与应用。企业数据中心基础设施架构正在面临着一场前所未有的变革，即从“硬件定义”向“软件定义”的云化基础设施架构演进。云计算将为运营商和企业带来业务敏捷性、核心生产力与竞争力的大幅提升，IT基础设施资源实现最优化配置。

那么，云计算新发展阶段具体的架构形态究竟应该是怎样的呢？是否存在一个对于所有垂直行业的企业数据中心基础设施云化演进，以及无论对于公有云和私有云场景都普遍适用的标准化云平台架构呢？

答案无疑是肯定的。就IT基础设施与上层软件应用的耦合度而言，在业务应用软件逻辑的执行层面上，由于Intel x86服务器架构已成为企业IT平台的普遍选择，以及x86服务器逐渐替代RISC及UNIX小型机，使得基于x86指令体系的二进制可执行代码成为普遍的选择。更进一步，由于Windows、Linux操作系统在各行业IT系统中的广泛采用与普及，操作系统层面的系统调用API也成为上层应用与服务器主机基础设施硬件资源交互的默认界面。这基本上完全解除了上层软件应用与底层硬件平台之间的耦合与依赖性。在业务应用软件与其所需的基础设施资源之间的管理调度层面上，通过对基础设施即服务（IaaS）面向上层的API定义的标准化与规范化，同样可以实现我们所期望的软硬件解耦。

从系统架构视角看，尽管私有云与公有云的外在商业模式与运营管理模式存在显著的差别，然而从技术视角来看，无论是公有云还是私有云，其核心实质是完全相同的：首先将硬件上分散的、孤立的多个设备资源，在逻辑上整合构建为一个大规模的统一资源池，然后再基于此资源池，以Web Portal或者API为界面，向外部云租户或者内部云租户提供按需分配与释放的基础设施资源池，云租户可以通过Web Portal或者API界面给出其从管理规划和应用需求视角出发对计算、存储、网络等基础设施资源的规模大小以及QoS/SLA量化规格方面的需求，并依赖云计算架构平台来实现对业务请求界面上所需的高度自动化的、弹性按需的资源供给。

综上所述，一套统一的云计算架构是完全可以同时覆盖于公有云和私有云应用场景的。

2.1.1 云计算核心架构上下文

云计算架构应用上下文的相关角色包括云租户/服务消费者、云应用开发者、云服务运营者/提供者、云设备提供者（见图2-1）。

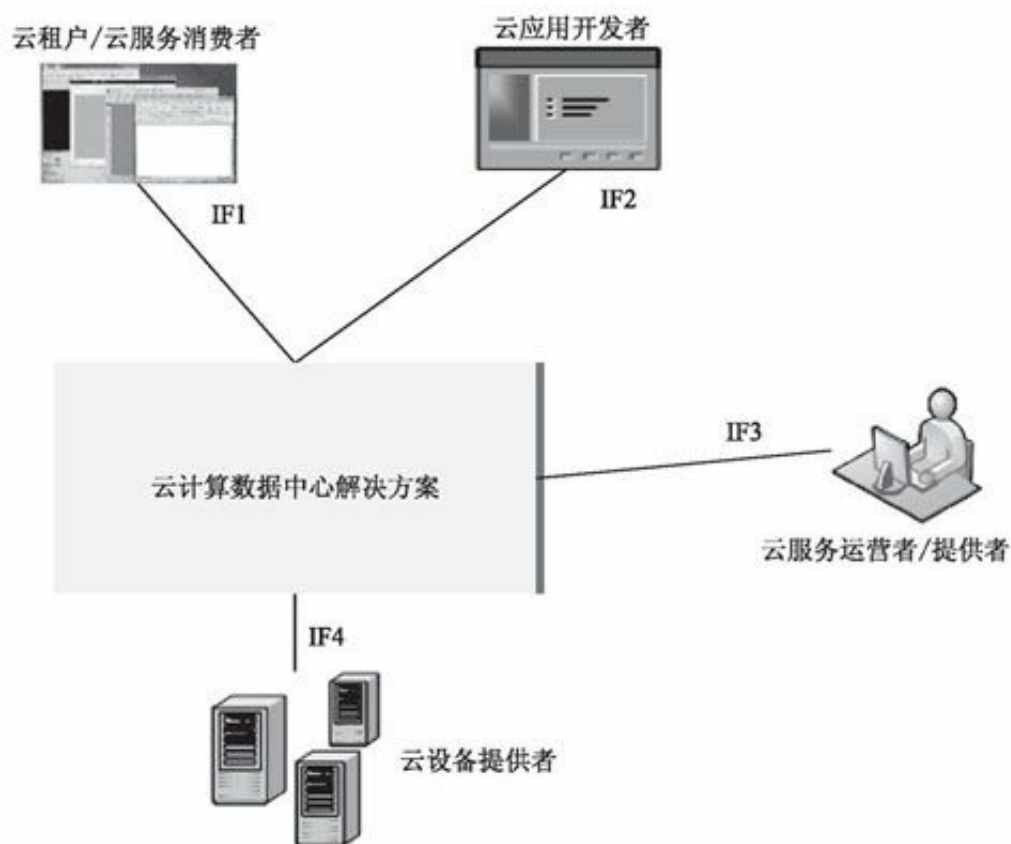


图2-1 云计算系统构架上下文

云租户/云服务消费者

云租户是指这样一类组织、个人或IT系统，该组织/个人/IT系统消费由云计算平台提供业务服务（比如请求使用云资源配额，改变指配给虚拟机的CPU处理能力，增加Web网站的并发处理能力等）。该云租户/云服务消费者可能会因其与云业务的交互而被计费。

云租户也可被看做是一个云租户/服务消费者组织的授权代表。比如说一个企业使用了云计算业务，该企业整体上相对云业务运营及提供者来说是业务消费者，但在该业务消费者内可能存在更多的细化角色，比如使得业务消费得以实施的技术人员，以及关注云业务消费财务方面的商务人员等。当然，在更为简化的公有云场景下，这些云业务消费者的角色关系将简化归并到一个角色。

云租户/服务消费者在自助Portal上浏览云服务货架上的服务目录，进行业务的初始化以及管理相关操作。

就多数云服务消费者而言，除从云服务提供者那里获取到的IT能力之外，还继续拥有其传统（非云计算模式）IT设施，这使得云服务与其内部既有的IT基础设施进行集成整合至关重要，因此特别需要在混合云的场景下引入云服务集成工具，以便实现既有IT设施与云服务之间的无缝集成、能力调用以及兼容互通。

云应用开发者

云应用开发者负责开发和创建一个云计算增值业务应用，该增值业务应用可以托管在云平台运营管理者环境内运行，或者由云租户（服务消费者）来运行。典型场景下云应用开发者依托于云平台的API能力进行增值业务的开发，但也可能会调用由BSS和OSS系统负责开放的云管理API能力（云应用开发者当然也可能选择独立构建其独立于云平台的增值业务应用系统的BSS/OSS系统，而不调用或重用底层的云管理API）。

云业务开发者全程负责云增值业务的设计、部署并维护运行时主体功能及其相关的管理功能。如同云租户/云业务消费者以及云业务运营提供者一样，云业务开发者也可以是一个组织或者个人，比如一个开发云业务的ISV开发商是一个云业务开发者，其内部可能包含了上百个担任不

同细分技术或商业角色的雇员。另外，负责云业务管理的运维管理人员与负责开发云业务的开发组织紧密集成也是一种常见的角色组织模式（比如Google、Amazon、百度等自营加自研的Internet DevOps服务商），这是提升云业务发放和上线效率的一种行之有效的措施，因为此类角色合一的模式提供了更短的问题反馈路径，使得云业务的运营效率有了进一步实质性提升。

目前云计算业务开发者在公有云及私有云领域的典型应用包括运营商虚拟主机出租与托管云、企业内部IT私有云或专有云、桌面私有云、运营商桌面云服务、企业网络存储与备份云、视频媒体处理云、IDC Web托管及CDN云以及大数据分析云等。

云服务运营者/提供者

云服务运营者/提供者承担着向云租户/服务消费者提供云服务的角色，云服务运营者/提供者概念的定义来源于其对OSS/BSS管理子系统拥有直接的或者虚拟的运营权。同时作为云服务运营者以及云服务消费者的个体，也可以成为其他对外转售云服务提供者的合作伙伴，消费其云服务，并在此基础上加入增值，并将增值后的云服务对外提供。当然，云服务运营者组织内部不排除有云业务开发者的可能性，这两类决策既可在同一组织内共存，也可相对独立进行。

云设备/物理基础设施提供者

云设备提供者提供各种物理设备，包括服务器、存储设备、网络设备、一体机设备，利用各种虚拟化平台，构筑成各种形式的云服务平台，这些云服务平台可能是某个地点的超大规模数据中心，也可能是由地理位置分布的区域数据中心组成的分布式云数据中心。

云设备提供者可能是云服务运营者/提供者，也可能就是一个纯粹的云设备提供者，他将云设备租用给云服务运营者/提供者。

在这里我们特别强调云设备/物理基础设施的提供者必须做到不与唯一的硬件设备厂家唯一绑定，即在云计算系统平台南向接口上所谓的多厂家硬件的异构能力。

接口说明

从上述云计算的基础上下文描述，我们不难看出这是一层介于上面IT应用层、中间件层以及传统数据中心管理控制层与下面数据中心物理基础设施层之间的一层软件。从宏观的数据中心资源池整合的视角来看，我们不妨称之为云操作系统（Cloud OS）。

云操作系统的南向接口IF4向下屏蔽底层千差万别的物理基础设施层硬件的厂家差异性。针对应用层软件以及管理软件所提出的基础设施资源诉求，云操作系统向上屏蔽如何在一个超大规模的逻辑资源池内进行调度协同的细节，并在北向接口IF1、IF2和IF3为上层软件及特定租户提供一个归一化、标准化的基础设施服务（IaaS）API服务接口。在云操作系统面向云运营和管理者的IF3接口之上，除了面向租户（拥有全局云资源操作权限）的基础设施资源生命周期管理API之外，还包括一些云租户API无法覆盖的，面向基础设施资源日常OAM的操作运维管理API接口。

其中IF1/IF2/IF3接口中关于云租户感知的基础设施资源生命周期管理API的典型形态为Web RESTFUL接口。IF4接口为业务应用执行平面的x86指令，以及基础设施硬件特有的、运行在物理主机特定类型OS中的管理Agent，或者基于SSL承载的OS命令行的管理连接。IF3接口中的OAM API则往往采用传统IT和电信网管中被广泛采用的Web RESTFUL、SNMP、CORBA等接口。

2.1.2 云计算平台架构

如本书前言所述，IT架构的演进经历了“合”、“分”、“合”三个阶段，如图2-2所示。

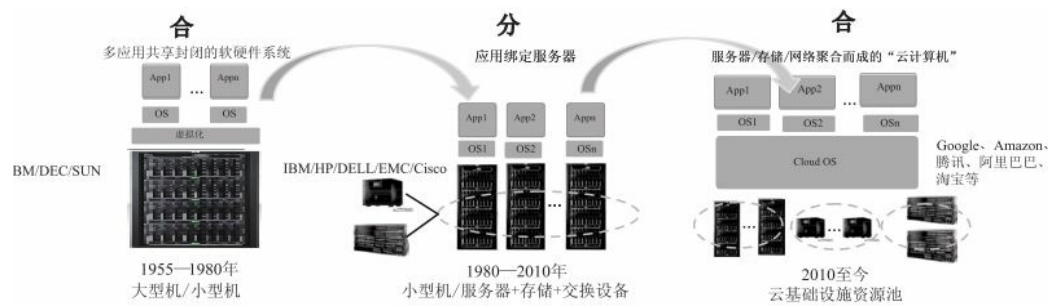


图2-2 IT架构演进路径

回顾企业IT架构演进的整个历史，我们不难看到，冯·诺依曼架构的第

一台计算机诞生以来的前30年，计算高度集中化、支持多用户多任务的大型机和小型机是企业IT的主流形态，构成IT系统的软硬件堆栈各层之间缺少统一的工业标准，呈现出内聚与耦合的特征，仅少数厂家拥有提供端到端高度复杂化的IT系统软硬件的能力。那个时代的IT系统造价高昂，往往是少数高端企业才能拥有的“奢侈品”。

于是，20世纪80年代，以x86服务器和PC系统的诞生为标志，企业IT系统迎来了第二次里程碑式的变革：从All in One、全封闭的软硬件栈走向水平分层的网络、存储、服务器、操作系统、中间件、应用层等多层次水平分工的架构，各层之间接口标准化、规范化，极大地简化了每一层的技术复杂度，各层IT产业链获得了大繁荣与大发展，涌现出一批优秀的专业化厂家，聚焦于提供该领域内质量最佳的产品和解决方案，IT系统终于开始走入“寻常百姓家”。

然而，所谓“物极必反”，当这个架构分层发展到一定阶段，弊端逐步显现。由于企业IT的层次太多，各层之间集成交付的难度越来越大，尤其是当今企业软件应用已从单一实例应用，迅速走向大规模分布式应用，一个关键业务的部署往往需要涉及服务器、网络、存储等各方面基础设施资源的协同配合，使得业务驱动的基础设施层服务器、存储、网络资源的集成管理配置和按需供给往往成为影响企业IT快速响应企业业务需求的关键制约因素。同时软硬件各层的开发虽然实现了解耦，但部署和运行态仍然是软硬件耦合绑定的关系，因此跨服务器的资源出现忙闲不均时，依旧无法有效利用IT资源。

随着企业信息化进程的不断推进，企业IT系统的使用者和维护者们逐渐发现，分层架构体系也存在着诸多弊端：

- 软硬件开发态解耦，但部署和运行态并未解耦；
- 生态链大繁荣的同时，多厂家硬件异构集成与管理的复杂度越来越高；
- 企业信息化的重心向软件转移，但计算、存储、网络硬件弹性供给能力及其相互协同的不足，越来越成为软件价值提升的制约性因素。

在IT架构演进的第三阶段，为实现云计算梦想，首先需要把所有IT基础

设施资源都高度集中化到一个数据中心去，通过云OS整合成为一台超大规模计算机，然后再按照每个租户的需求将资源动态切分提供给每个用户。于是从总体架构上我们发现了一个有趣的现象，我们的IT基础设施架构经历了一个从合、到分、再到合的过程。大型机时代，很多应用实例跑在封闭的平台之上，用户只能通过哑终端去访问集中的计算资源。到20世纪80年代，应用分布在多个处理能力相对较弱的处理单元上。进入当前的云计算年代，计算资源又重新回归整合，只不过不是封闭的硬件整合，而是基于松耦合独立节点网络连接，以及统一逻辑调度管控的整合。

“天下大势，合久必分，分久必合”。IT基础设施架构从合并、分离并重新走向融合，借助虚拟化及分布式云计算调度管理软件，将IT基础设施整合成为一个规模超大的“云计算机”，相当于建成了一座“基础设施电厂”。多个租户可以从这座电厂中随时随地获取到其所需的资源。云计算大大提升了业务敏捷度，降低了TCO消耗，甚至提供了更优的业务性能与用户体验。

然而，历史总是在螺旋式前进的，“云计算机”看似大型机，但绝非简单回到了大型机时代。

➤ 区别1：架构不同，规模扩展能力不同

由“垂直扩展”到“横向扩展”：计算处理能力、存储容量、网络吞吐能力、租户/应用实例数量均相差 n 个数量级以上，从硬件成本视角来看，TCO成本也较低。

➤ 区别2：硬件依赖性不同，生态链、开放性不同

由“硬件定义”到“软件定义”：早期的IT系统，少数硬件厂家绑定OS和软件，IT只是少数用户的奢侈品。在新的时代，通过软件屏蔽异构硬件差异性，在同一个硬件平台上可以运行来自多个不同厂家的软件和OS。新时代的IT生态链更加繁荣，IT成为人人消费得起的日用品。

➤ 区别3：可靠性保障方式不同

其由“单机硬件器件级的冗余实现可靠性”发展到“依赖分布式软件和故障处理自动化实现可靠性（甚至支持地理级容灾）”。

➤ 区别4：资源接入方式不同

将IT基础设施能力比做“电力”，大型机只能专线接入，是只能服务于少数人群的“发电机”，基于企业以太网或者互联网的开放接入，可以为更多的人群提供服务的“发电站”和“配电网”。

基于上述分析，我们不难得出如图2-3所示的云计算数据中心架构分层概要。

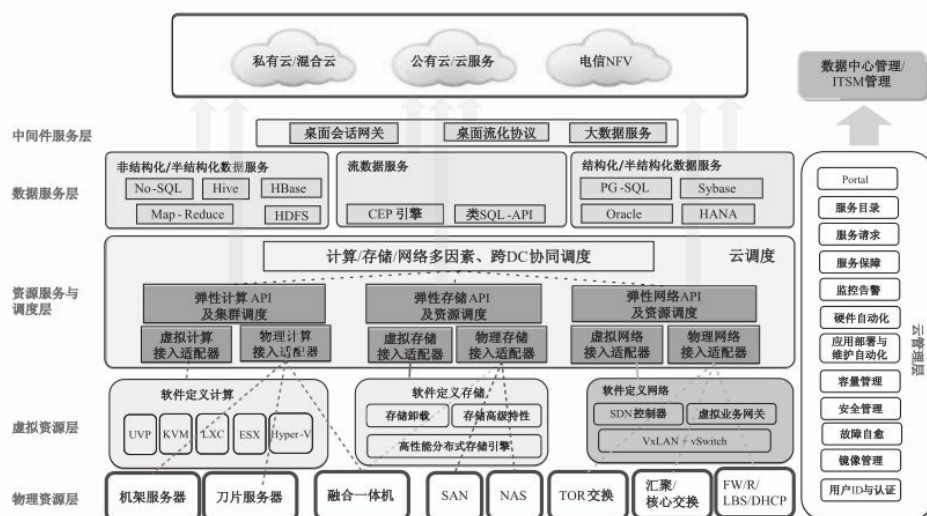


图2-3 云计算数据中心解决方案端到端总体分层架构

物理资源层

所有支撑IaaS层的IT基础设施硬件包括服务器、存储（传统RAID架构垂直扩展的Scale-up存储，以及基于服务器的分布式水平扩展的Scale-out存储）、数据中心交换机（柜顶、汇聚以及核心交换）、防火墙、VPN网关、路由器等网络安全设备。

虚拟资源层

虚拟资源层在云计算架构中处于最为关键与核心位置。该层次与“资源服务与调度层”一道，通过对来自上层操作系统及应用程序对各类数据中心基础设施在业务执行和数据平面上的资源访问指令进行“截获”。指令和数据被截获后进行“小聚大”的分布式资源聚合处理，“大分小”的虚拟化隔离处理，以及必要异构资源适配处理。这种处理可以实现在上层操作系统及应用程序基本无需感知的情况下，将分散在一个或多个数据

中心的数据中心基础设施资源统一虚拟化与池化。

在某种程度上，对于应用程序的支撑关系，虚拟资源层对于上层虚拟机（含操作系统及应用程序）的作用与操作系统是类似的，实质上都是在多道应用作业实例与底层的物理资源设备或者设备集群之间进行时分和空分的调度，从而让每道作业实例都“感觉”到自己是在独占相关资源，而实际上资源在多个作业实例之间的复杂、动态的复用调度机制完全被虚拟资源层屏蔽。技术实现的主要困难与挑战在于，操作系统的管理API是应用程序感知的，而虚拟资源层则必须做到上层操作系统与应用程序的“无感知”，同时对于频繁的指令级陷入和仿真调度助力，做到令上层应用及OS可接受的性能开销。

虚拟资源层包括三个部分，具体如下。

（1）计算虚拟化

所有计算应用（含OS）并非直接承载在硬件平台上，而是在上层软件与裸机硬件之间插入一层弹性计算资源管理及虚拟化软件：弹性计算资源管理软件对外负责提供弹性计算资源服务管理API，对内负责根据用户请求调度分配具体的物理机资源；虚拟化软件（Hypervisor）对所有的x86指令进行截获，并执行不为上层软件（含OS）所知的多道执行环境并行的“仿真操作”，使得从每个上层软件实例的视角来看，其仍然独占底层的CPU、内存以及I/O资源（见图2-4）；而从虚拟化软件的视角来看，则是将裸机硬件在多个客户机（VM）之间进行时间和空间维度的穿插共享（时间片调度，I/O多队列模拟等）。由此可见，计算虚拟化引擎本身是一层介于OS与硬件平台中间的附加软件层，因此将不可避免地带来性能上的损耗。然而随着云计算规模商用阶段的到来，以及计算虚拟化的进一步广泛普及应用，越来越多的计算性能敏感型和事务型的应用逐步从物理机平台迁移到虚拟化平台之上，因此对进一步降低计算虚拟化层的性能开销提出了更高的要求，典型的增强技术包括以下内容。

➤ 虚拟化环境下更高的内存访问效率：应用感知的大内存业务映射技术，通过该技术，可有效提升从虚拟机线性逻辑地址到最终物理地址的映射效率。

➤ 虚拟化环境下更高的CPU指令执行效率：通过对机器码指令执行的流程进行优化扫描，通过将相邻执行代码段中的“特权”指令所

触发的“VM_Exit”虚拟化仿真操作进行基于等效操作的“合并”，从内容达到在短时间内被频繁反复地执行。由于每次VM_Exit上下文进入和退出的过程都需要涉及系统运行队列调度以及运行环境的保存和恢复，即将多次上下文切换合并为一次切换，从而达到提升运行效率的目的。

➤ 虚拟化环境下更高的I/O和网络包收发处理效率：多个虚拟机在一个物理机内需要共享相同的物理网卡进行网络包收发处理，从而有效减少中断处理带来的开销；在网络及I/O发包过程中，通过将小尺寸分组包合并为更大尺寸的分组包，可以减少网络收发接受端的中断次数，从而达到提升虚拟机之间网络吞吐率的目的。

➤ 更高的RAS可靠性保障：针对云计算所面临的电信领域网络及业务云化的场景，由于硬件故障被虚拟化层屏蔽了，使得物理硬件的故障无法像在传统物理机运行环境那样直接被传送通知给上层业务软件，从而导致上层业务层无法对故障做出秒级以内的及时响应，比如业务层的倒换控制，从而降低了对整体的可靠性水平。如何感知上层的业务要求，快速进行故障检测和故障恢复，保证业务不中断，这给计算虚拟化带来了新的挑战。

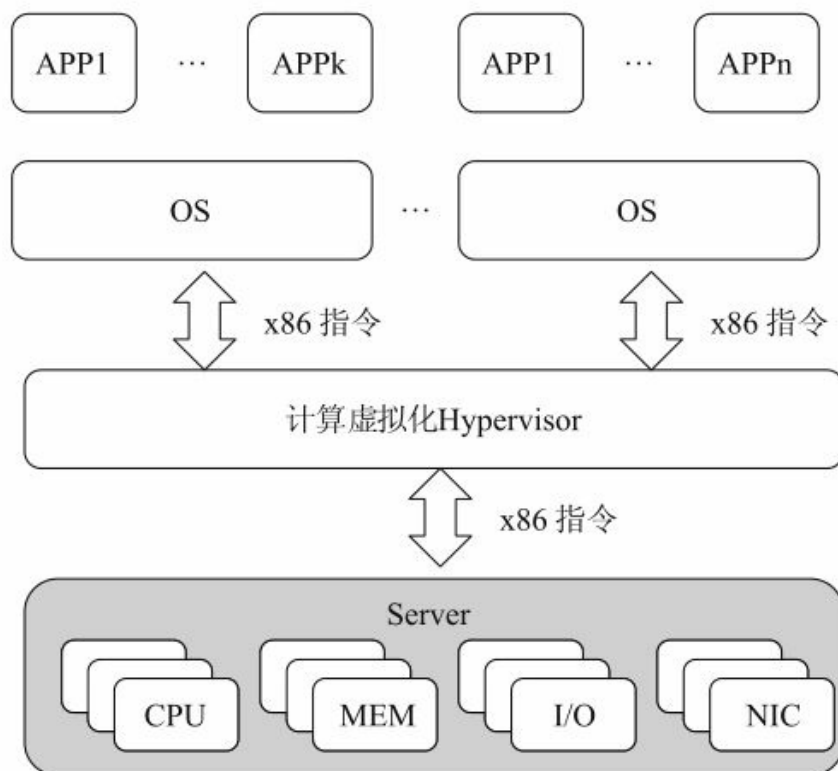


图2-4 计算虚拟化硬件接口

（2）存储虚拟化

随着计算虚拟化在各行业数据中心的普遍采用，x86服务器利用效率已得到提升，人们发现，存储资源的多厂家异构管理复杂，平均资源利用效率低下，甚至在I/O吞吐性能方面无法有效支撑企业关键事务及分析应用对存储性能提出的挑战，通过对所有来自应用软件层的存储数据面的I/O读写操作指令进行“截获”，建立从业务应用视角覆盖不同厂家、不同版本的异构硬件资源的统一的API接口，进行统一的信息建模，使得上层应用软件可以采用规范一致的、与底层具体硬件内部实现细节解耦的方式访问底层存储资源。

除去带来硬件异构、应用软件与硬件平台解耦的价值外，其还可以通过“存储虚拟化”层内对多个对等的分布式资源节点的聚合，实现该资源的“小聚大”。比如，将多个存储/硬盘整合成为一个容量可无极扩展的超大（EB级规模）的共享存储资源池。由此可以看到，存储虚拟化相对计算虚拟化最大的差别在于：其主要定位是进行资源的“小聚大”，而非“大分小”。原因在于，存储资源的“大分小”在单机和SAN/NAS独立存储系统存储，乃至在文件系统中通过LUN划分及卷配置已经天然实现了，然而随着企业IT与业务数据的爆炸式增长，需要实现高度扁平化、归一化和连续空间，跨越多个厂家服务器及存储设备的数据中心级统一存储，即“小聚大”。存储“小聚大”的整合正在日益凸显出其不可替代的关键价值（见图2-5）。

➤ 高性能分布式存储引擎：伴随着云计算系统支撑的IT系统越来越大，覆盖范围从不同服务器存储节点，到分布在不同地理区域的数据中心，这就需要有一个分布式存储引擎。这个引擎，能满足高带宽、高I/O等各种场景要求，能很好地进行带宽的扩展。

➤ 存储异构能力：如何利旧，将不同厂家原有的独立SAN、NAS设备组合成一个大的存储资源池，也是软件定义存储中需要解决的问题。

➤ 存储卸载：传统的企业存储系统采用各种各样的存储软件，这些软件存储操作对存储I/O和CPU资源均有较大消耗，会影响用户业务性能的发挥。因此如何将存储操作标准化，然后将存储操作利

用某些标准的硬件动作去代替，这就是存储卸载。

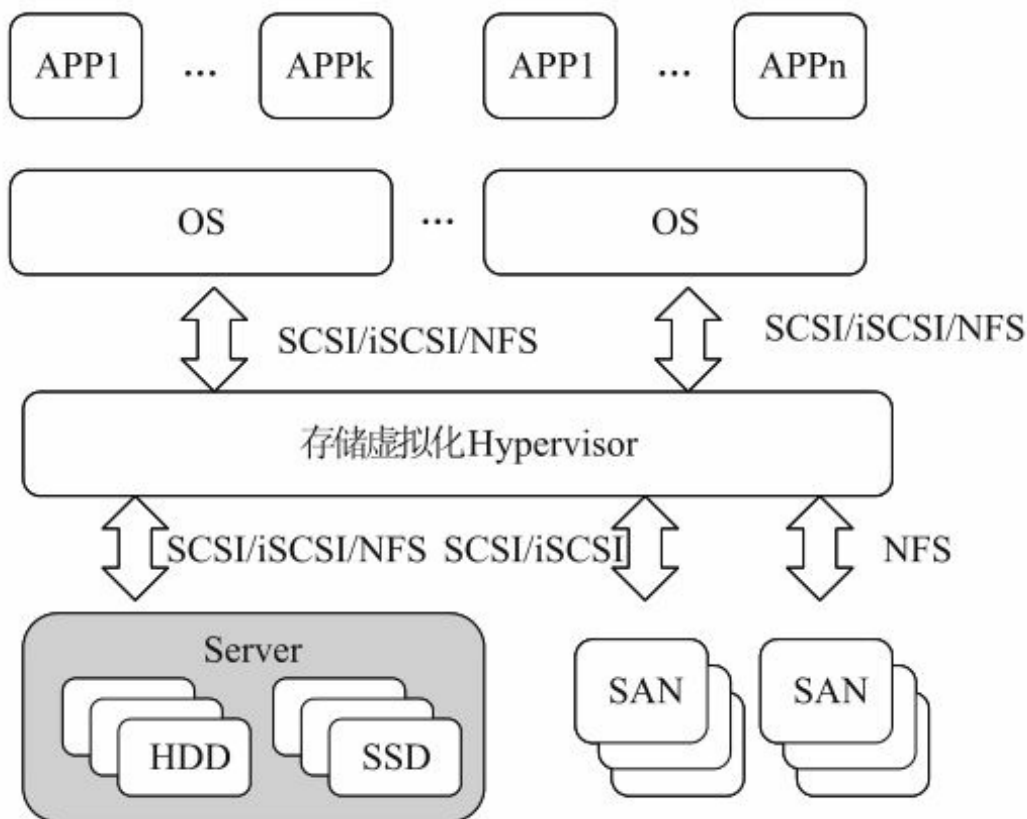


图2-5 存储虚拟化硬件接口

(3) 网络虚拟化

站在操作系统角度，OS管理的资源范畴仅仅是一台服务器，而Cloud OS管理的资源范畴扩展到了整个数据中心，甚至将跨越多个由广域网物理或者逻辑专线连接起来的数据中心。在一台服务器内，核心CPU、内存计算单元与周边I/O单元的连接一般通过PCI总线以主从控制的方式来完成，多数管理细节被Intel CPU硬件及主板厂家的总线驱动所屏蔽，且PCI I/O设备的数量和种类有限，因此OS软件层面对于I/O设备的管理是比较简单的。相对而言，在一个具备一定规模的数据中心内，甚至多个数据中心，各计算、存储单元之间以完全点对点的方式进行松耦合的网络互联。云数据中心之上承载的业务种类众多，各业务类型对于不同计算单元（物理机、虚拟机）之间，计算单元与存储单元之间，乃至不同安全层次的计算单元与外部开放互联网和内部企业网络之间的安全隔离及防护机制，要求动态实现不同云租户之间的安全隔离。云数据中心还要满足不同终端用户不同场景的业务组网要求以及他们的安全隔离要

求。因此，云操作系统的复杂性将随着云租户及租户内物理机和虚拟机实例的数量增长呈现几何级数的增长，由业务应用驱动的数据中心网络虚拟化和自动化就变得势在必行和不可或缺。为了实现彻底与现有物理硬件网络解耦的网络虚拟化与自动化，唯一的途径与解决方案就是SDN（即所谓软件定义的网络），即构建出一个与物理网络完全独立的叠加式逻辑网络，其主要部件以及相关技术包括以下几个方面。

- **SDN控制器：**这是软件定义网络的集中控制模块，负责云系统中网络资源的自动发现和池化，根据用户需求分配网络资源，控制云系统中网络资源的正常运行。
- **虚拟交换机：**根据SDN控制器创建出虚拟交换机实例。可以对这个虚拟交换机进行组网的设计、参数的设置，一如对物理交换机的使用。
- **虚拟路由器：**根据SDN控制器创建出虚拟路由器实例。可以对这个虚拟路由器进行组网的设计、参数的设置，一如对物理路由器的使用。
- **虚拟业务网关：**根据用户业务的申请，由SDN控制器创建出的虚拟业务网关实例，提供虚拟防火墙的功能。可以对这个虚拟业务网关进行组网的设计、参数的设置，一如对物理业务网关的使用。
- **虚拟网络建模：**面对如此复杂多变的组网，如何既保证网络的有效区分和管理，又保证交换和路由的效率，此时需要一个有效的建模方法和评估模型；虚拟网络建模技术能提前预知一个虚拟网络的运行消耗、效率 and 安全性。虚拟网络建模可以做成一个独立功能库，在需要的时候启动，以减少对系统资源的占用。

资源服务与调度层

相对虚拟化层在业务执行面和数据面上“资源聚合与分割仿真”，该层次主要体现为管理平面上的“逻辑资源调度”。

由于多个厂家已经投入到云计算的研发和实施中，不可避免地有多种实现方式；而要实现云计算真正的产业化并被广泛使用，各厂家的云计算平台必须能够互相交互，即进行接口标准化。接口标准化后，主流的虚

拟化平台（例如Hyper-V、KVM、UVP、ESX等）之间能够互相兼容，各个硬件厂家或者中间件厂家可以自由选择虚拟化内核。

在云计算新的发展阶段中，面向公有云、面对国际化公司的分布式云系统将是重点，这将引发对超大资源的分配和调度。在整个云计算的实现架构上，计算、存储、网络资源的分配和使用将走向专业化。这是因为根据性质的不同，云应用业务对计算、存储、网络资源的需求可能是不一样的，例如对于呼叫中心业务，其偏向于计算资源使用，而对于网盘业务，则偏向于存储资源使用。在这种情况下，为了更有效地利用资源，给业务层提供基本资源调用API是最好的选择，将计算、存储、网络资源作为基本资源单位，提供统一的资源调用接口，让云业务开发者自己选择如何高效地使用这些资源。这些API包括以下几个方面。

➤ **弹性计算资源调用API：**计算资源包括CPU和内存，云计算平台根据云运营商的要求，已经将CPU和内存虚拟化和池化。系统提供资源的动态申请、释放、故障检测、隔离和自动切换功能，做到业务不感知。CPU资源又可以分为纯计算型、图像处理型等不同类型；不管是CPU还是内存，都提供瘦分配功能，资源的自动伸缩保证在低业务量时减少资源的消耗，高业务量时开启所有物理资源，确认业务的高效运行。计算资源API还需要提供集群能力。

➤ **弹性存储资源调用API：**存储资源API提供文件或者卷接口，除了提供常见的资源申请、释放、瘦分配等功能外，还涉及其他几个关键方面。

- **异构资源的池化：**不同的厂家在将存储资源池化后，提供统一的API，一个厂家可以利用这些API，将不同厂家的存储资源池构成一个大的资源池，然后再封装出API供业务调用。
- **存储资源的分层分级存储：**因业务性能要求的不同，分层存储是一个常用的技术，业务系统在申请存储资源的时候，可以选择是否使用这个特性。
- **内存存储资源的支持：**未来的系统，内存一定会成为主存，所有的存储，除非一些特别重要的信息，基本上不再需要存入非易失性介质；而使用内存资源作为主存，可靠性是关键要求；在构造内存存储池的时候，可靠性必须贯彻始终，确保每个内存存储有备份，或者确保内存存储有可靠的UPS保护。

➤ **弹性网络资源调用API：**网络资源API的基本功能包括资源的申请、释放、监控、故障隔离和恢复等，需要考虑异构资源的统一化。

数据服务层

数据服务层是叠加在基础设施服务之上的，具备多租户感知能力的结构化、半结构化及非结构化数据服务能力。

通过对弹性资源层和数据服务层的一些紧密接口的实现，提高数据存储的效率。

➤ **结构化数据服务：**结构化数据服务子层提供对结构化数据的存储和处理功能，它通过叠加各种结构化数据库软件来实现，例如常见的Oracle\Sybase\HANA等。为提高处理效率，弹性存储资源调度层会针对不同的基于磁盘或者基于内存的数据库，提供更高效的存储资源调用API，例如面向HANA内存数据库，提供内存专用的存储资源调用API接口。

➤ **非结构化数据服务：**这个子层主要是叠加常见的No SQL数据库的功能模块，例如Map-reduce、HBase等，提供弹性存储资源的特殊接口。

➤ **流数据服务：**流数据服务更多地涉及对特殊CPU资源和专用芯片资源的使用，在弹性计算资源API中提供一些专用接口，来进行流数据的高效输入、压缩/解压缩、处理和转发。

云管理层

除掉上面各子系统之外，云管理层还有纵向拉通云平台各服务层及子系统的云管理子系统，负责端到端云计算服务实例的创建发放，生命周期管理，服务SLA水平保障，云计算数据中心物理及虚拟化平台基础设施，以及平台基础设施与上层业务关联的FCAPS日常操作与维护业务。

云管理与传统电信的OSS/BSS系统的最大差异在于其多种不同横向与纵向资源整合的全自动化、智能化的支持。

中间件服务层

为提供一些基础服务，系统在某些场景下还需要做一些适配服务，例如，提供桌面云时，需要提供桌面云相关的应用协议、桌面应用的调度等；面向不同业务群提供大数据服务时，需要做一些应用的适配。

2.2 云计算架构的关键技术

相对于云计算初级阶段以探索和试用为特征的非互联网领域及行业的基础设施云资源池建设，新阶段云计算基础设施云化已步入大规模集中化建设的阶段，需要云操作系统（Cloud OS）必须具备对多地多数据中心内异构多厂家的计算、存储以及网络资源的全面整合能力，因此有如下一些关键技术和算法。

2.2.1 超大规模资源调度算法

我们说希望能像用水用电一样的方式去使用IT资源，那么IT资源的供给就需要许多类似于大大小小的水厂/电厂的IT资源工厂，这就是我们所说的IT数据中心。

以水厂为例，其实我们有各种大小的水库，有时候为了供给一个大城市，我们会通过复杂的管道，将某些江河或某些水库引入城市边上的大水库。就是说，这个供水系统是一个复杂的网络系统，需要有良好的预先设计。

可以看到，尽管我们家家户户使用的自来水没有什么差别，但实际上他们是来自于不同的水厂。每个水厂都可能遇到自己的枯水期，使用这些水厂水资源的客户可能面临缺水的问题，就是说，水的供应并不是无限制的。对应地，其实我们需要的IT资源也并不是无限制供给的，它是由后面大大小小的IT数据中心的能力决定的。当然，为了应对一些大企业的IT资源要求，我们需要将异地的IT数据中心进行联网设计，组成一个大的IT资源池来给大客户使用；此时，这个大资源池的组成技术、调度技术都是关键技术，包括以下三个方面。

1. 计算资源调度算法

超大规模资源调度算法实现十万物理机、百万虚拟机的多级、分层调度。

$$\forall i,h \quad e_{ih} \in \{0,1\} \quad y_{ih} \in Q \quad (1)$$

$$\forall i \quad \sum_h e_{ih} = 1 \quad (2)$$

$$\forall i,h \quad 0 \leq y_{ih} \leq e_{ih} \quad (3)$$

$$\forall i \quad \sum_h y_{ih} \geq \hat{y}_i \quad (4)$$

$$\forall h,j \quad \sum_i r_{ij}(y_{ih}(1-\delta_{ij})+e_{ih}\delta_{ij}) \leq 1 \quad (5)$$

$$\forall i \quad \sum_h y_{ih} \geq \hat{y}_i + Y(1-\hat{y}_i) \quad (6)$$

图2-7 资源弹性分配

其中各变量的含义如下。

$i=1.....N$ ，表示服务请求数量。

$h=1.....H$ ，表示每个集群中同质物理服务器的数量。

$j=1.....d$ ，表示每个服务器提供的资源类型数量（例如CPU、RAM、带宽等）。

R_{ij} 表示第*i*个服务请求对资源类型*j*的资源需求量，这个值在0和1之间，表示资源的满足程度。

δ_{ij} 表示 R_{ij} 是否为固定资源请求类型，取值0或者1。如果 R_{ij} 是固定资源请求（比如每个用户邮箱服务固定需要内存10G），则 $\delta_{ij}=1$ ；如果 R_{ij} 是弹性资源请求（比如每个用户邮箱服务需要的内存可以在0~10G之间），则 $\delta_{ij}=0$ 。

\hat{y}_i ，表示服务*i*的最小产出要求，取值在0和1之间。例如某个大型企业需要从云中获得邮箱服务1 000个用户，并且客户要求无论如何，最差的情况下也需要保证服务200个用户，则此时取值0.2。

e_{ih} 表示资源请求*i*是否分配在物理服务器*h*上，取值0或者1。如果是，则

取值1；否则取值0。

y_{ih} 表示服务i在服务器h上是否进行Scale方式的输出。如果服务i不在这个服务器上，则取值必须是0。

各个限定表达式的含义具体如下。

表达式（1）：表示服务请求i分配在物理服务器h上的状态，或者在或者不在。

表达式（2）：表示无论某个服务器是否承载了服务请求i，所有服务器上满足服务请求i的总和肯定等于1。

表达式（3）：表示一个服务i可以在某个服务器h上得到部分运行资源的满足，这个满足程度肯定大于等于0，如果大于0而小于1，表示服务器i需要的资源是分配在多个服务器上的，此服务器只能满足部分资源需求；如果等于1，表示此服务此时无需进行Scale，它完全能在h服务器上得到全部的资源满足。

表达式（4）：表示一个服务在相关的服务器上能取得的产出必须大于它的最小产出需求。例如客户要求云系统满足最低200个邮箱用户的需求，而系统中有10万台服务器，不管此时有多少台服务器给这个企业客户提供邮箱服务，都必须保证200个邮箱用户的使用。

表达式（5）：表示资源类型j在服务器h上能分配各种服务使用的最大值是1。

表达式（6）：表示最小的服务产出Y不会大于任何服务的产出。

资源弹性分配的近似最优解有如下公式：

$$Y = \min \left(1, \min_{j \in NZ} \frac{H - \sum_i r_{ij} (\hat{y}_i (1 - \delta_{ij}) + \delta_{ij})}{\sum_i (1 - \hat{y}_i) r_{ij} (1 - \delta_{ij})} \right)$$

其中NZ表示不等于0的物理资源集合。

整个系统的求解就是获得Y值的最大值。一般来说，我们可以采用如下

的条件来获得最优解：

$$e_{ih}=1/H \text{ and } y_{ih}=\frac{1}{H}(\hat{y}_i+Y(1-\hat{y}_i))$$

2. 存储资源调度算法

存储资源的调度算法主要实现以下几点（见图2-8）。

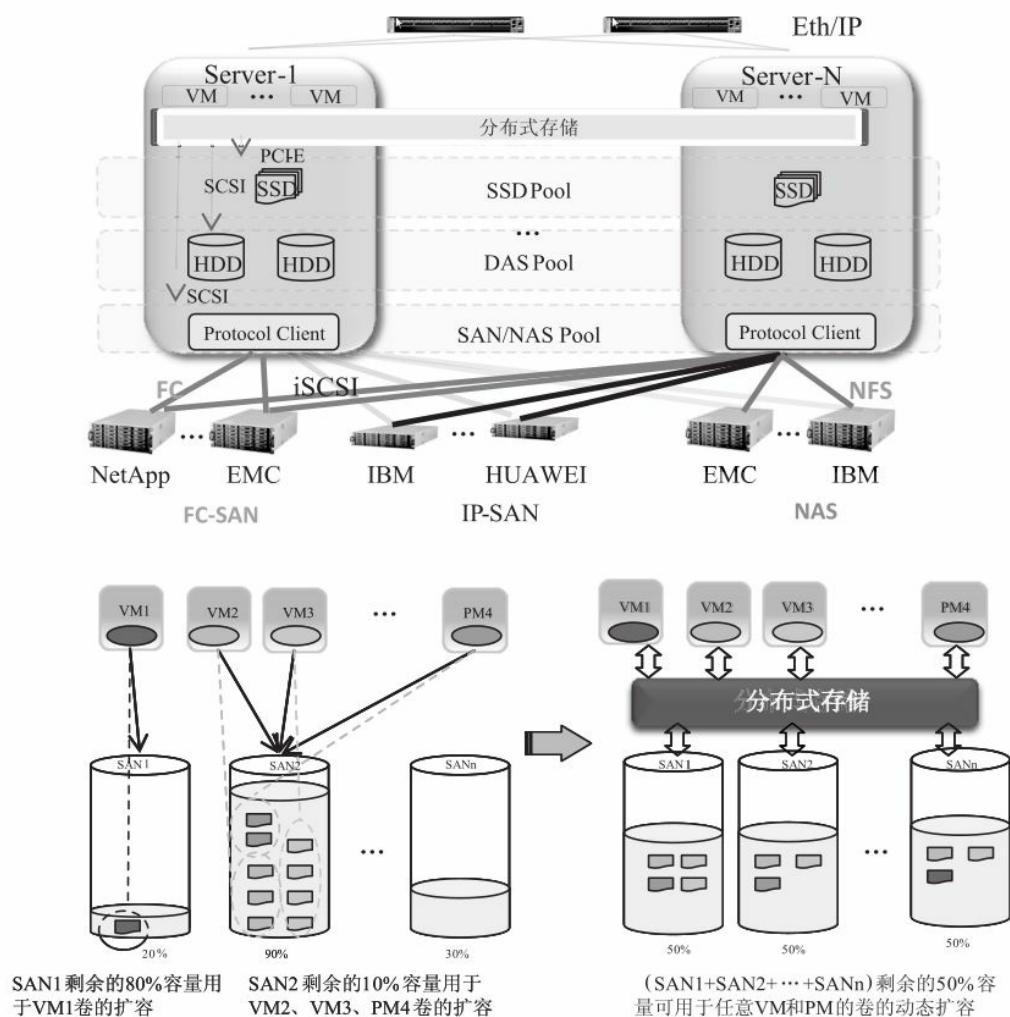


图2-8 异构大存储池

➤ 将数据中心服务器（机架式服务器）直连存储（HDD/SSD）转换为高性能、低时延的共享存储资源，大幅提升可用存储空间，实现无SAN化的计算集群的虚拟化整合；

- 为每VM/PM提供更大的“瘦分配弹性”，为不具备“瘦分配”能力的服务器内置DAS/SSD以及外置SAN带来天然瘦分配能力，并解决多数外置SAN存储瘦分配带来的性能下降开销问题；
- 更大规模的跨SAN资源池，基于在线分布式去重实现更大范围的重复数据识别与删除（文件级/对象级/块级），将资源利用率进一步提升40%；
- 更大规模的资源池，意味着可有更多共享空闲资源满足计算侧需求，避免独立SAN/NAS数据不均衡带来的资源浪费（30%）；
- 在超大资源池下，将“跨SAN”数据热迁移的概率几乎降低为零；
- 物理机无Hypervisor，需要引入“存储融合”层来解决数据的跨SAN热迁移能力（存储大资源池内的）。

3. 能耗管理最优化算法

要降低PUE值，实现云计算绿色节能的理念，需要有一个好的能耗管理算法。能耗管理算法在云计算中是一个关键技术（见图2-9）。

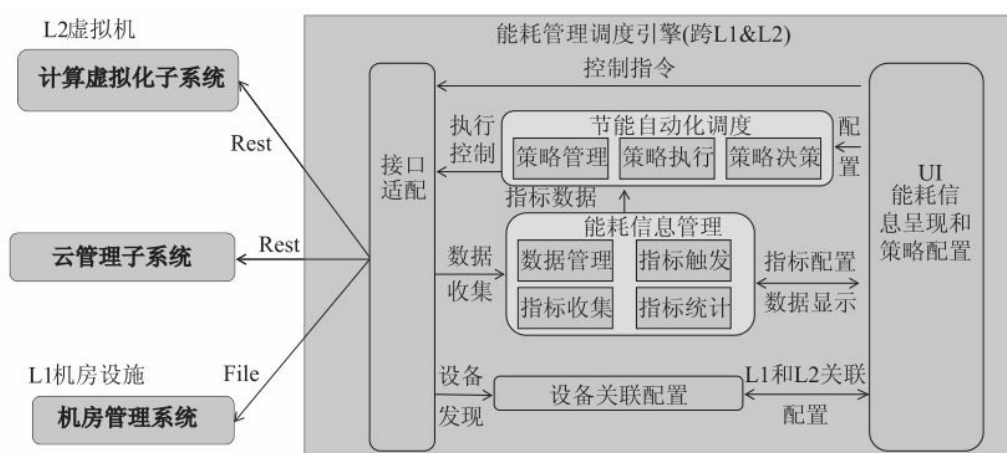


图2-9 能耗管理功能模块图

- 计算部分是数据中心L1+L2功耗的主要矛盾和关键路径（60%~70%）；

- 数据中心内，基于“轻载合并”原则进行VM热迁移调度，使得更多的空闲服务器可以下电或处于节能运行态；
- 计算虚拟化部分与数据中心L1管理软件联动，尽量减少局部热点，从而允许L1管理软件控制空调提升平均工作温度，达到提升PUE效率的目的；
- 持续动态采集当前负载情况下服务器、UPS及空调、制冷设备的功耗及温度数据，得出PUE指标，并在管理界面上实时呈现。

处理过程具体如下。

（1）输入信息

1)物理机信息列表

- 静态规格：CPU主频和数量、内存大小、网卡速率。
- 负载信息：CPU利用率、内存利用率、网络I/O。
- 状态：上电、下电、异常状态。
- 功率信息（可选）：额定功耗、当前功率。
- 温度信息（可选）：当前CPU温度、物理机温度。
- 其他（可选）：物理机能耗效率评级、离冷风送风口距离或评级。

2)虚拟机信息列表

- 静态规格：虚拟机CPU（vCPU数量和主频）、内存大小、网卡速率。
- 负载信息：CPU占用、内存占用、网络I/O。

- 约束信息：互斥性约束、亲和性约束。
- 物理机和虚拟机的关联关系。
- 物理机对应的VM ID列表。

3)两个场景

- 轻载时，合并VM，物理机下电节能。
- 重载时，启动物理机，均衡VM，保证QoS。

4)三个子算法

- 轻载/重载检测算法。
- 上下电PM选择子算法。
- 负载均衡子算法。

5)算法设计时要考虑的问题

- 多维资源问题。
- 迁移成本-收益分析。
- 迁移震荡问题。
- What-if测试。
- 配电问题。
- 温度问题。
- 调度约束。

（2）输出信息

其主要有两种动作，即物理机上下电动作、VM迁移动作。

2.2.2 异构集成技术

1. 异构硬件简化管理集成技术

异构的内容包括以下几点。

- Hypervisor层硬件异构：例如同时支持UVP & KVM引擎，广泛兼容其他厂家x86服务器。
- 硬件OM管理异构：云管理抽象出与设备无关的对象模型（CIM），如通过适配包采集模型中需要的数据，屏蔽不同硬件之间差异。适配包独立于主干版本发布，可动态加载到系统中以满足快速适配新硬件的能力。
- 虚拟机、物理机统一建模：x86服务器虚拟机、物理机，以及ARM物理机的异构集群管理。

异构实现原理如图2-10所示。

业务运行平面上，依托从Linux OS衍生出来的XEN与KVM虚拟化引擎，实现屏蔽硬件差异化的虚拟硬件仿真，对异构硬件特有的I/O驱动进行版本化验证管理。

管理平台上，在云管理子系统中采用灵活的插件机制对各类异构硬件通过有代理以及无代理模式的管理，从各类服务器硬件管理总线以及操作系统内的Agent，甚至异构硬件自带的管理系统中收集，并适配到统一的CIM信息模型中来。

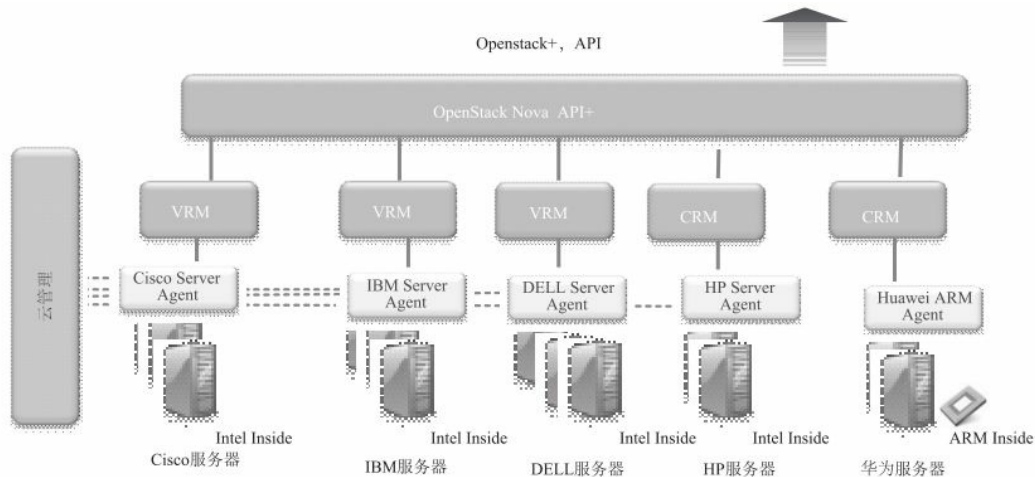


图2-10 硬件异构兼容原理

2. 异构Hypervisor简化管理集成技术

针对数据中心场景，企业IT系统中的Hypervisor选择往往不是唯一的，可能有VMware的ESX主机及vSphere集群，可能有微软的Hyper-V及SystemCenter集群，也可能有从开源KVM/XEN衍生的Hypervisor（如华为UVP等），多种选择并存。此时云操作系统是否有能力对这些异构Hypervisor加以统一调度管理呢？答案是肯定的。可以依托Openstack开源框架，通过Plug-in及Driver等扩展机制，将业界所有主流的Hypervisor主机或者主机集群管理接口统一适配到OpenStack的信息模型中来，并提供V2V/P2V虚拟机镜像的转换工具，在异构Hypervisor之间按需进行虚拟机镜像转换。这样即使不同的Hypervisor也可共存于同一集群，共享相同存储及网络服务，甚至HA服务。

资源以统一集群方式管理（OpenStack目标），屏蔽Hypervisor差异，简化云计算资源管理（见图2-11）。

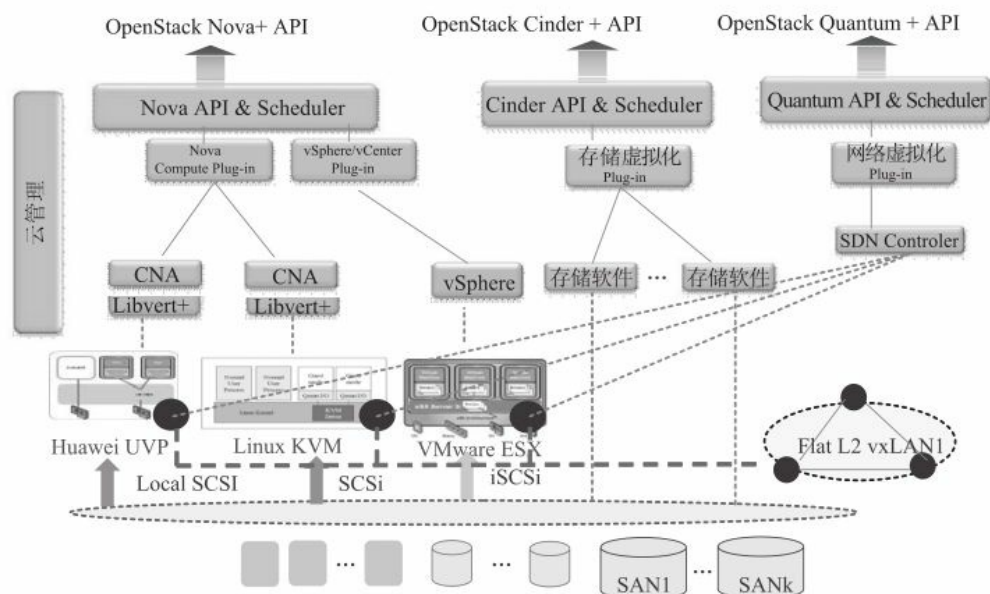


图2-11 Hypervisor异构统一管理原理

3. 异构存储管理集成的统一简化技术

异构存储管理继承的统一简化技术主要包括如下几点（见图2-12）。

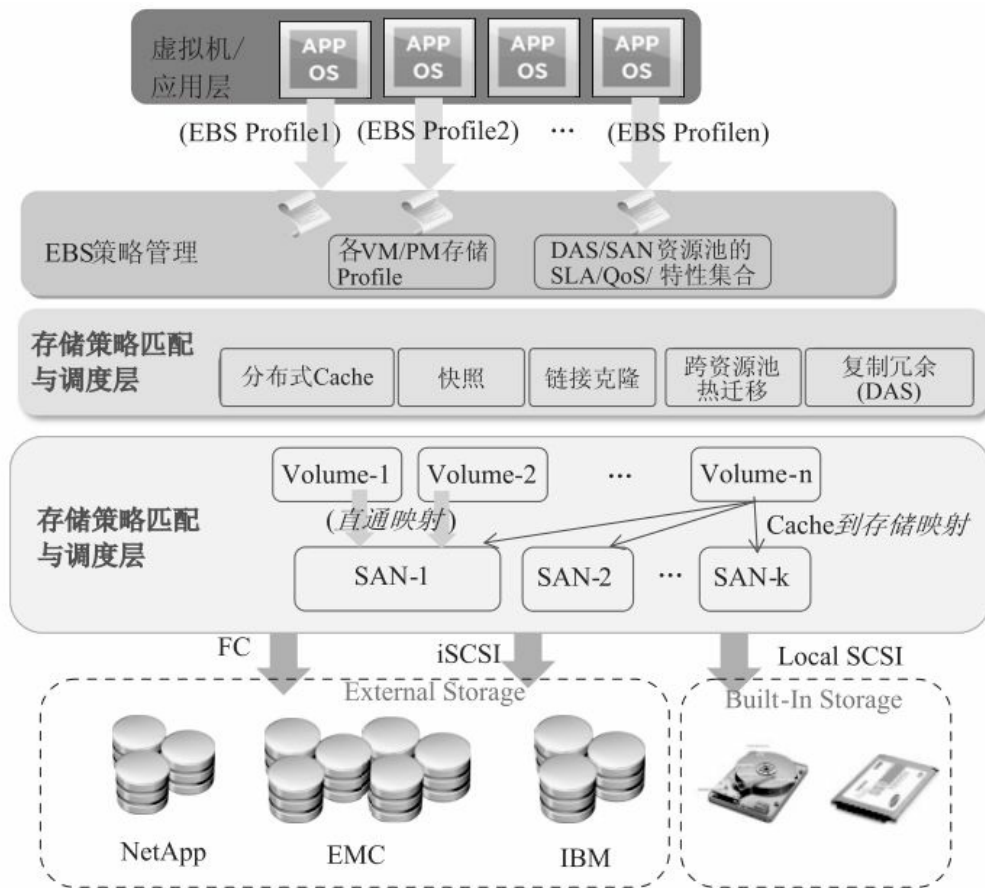


图2-12 存储异构统一管理原理

- 10PB级存储大资源池，跨多厂家异构外置存储，以及服务器自带SSD/HDD的资源池化，将存储服务抽象为同时适用于虚拟机和物理机的“统一EBS”服务；
- 容量、IOPS、MBPS等SLA/QoS是EBS存储服务界面的“统一语言”，与具体支撑该服务的存储形态及厂家无关；
- 可按需将部分存储高级功能（数据冗余保护、置0操作、内部LUN拷贝、链接克隆等）卸载到外置存储（类VVOL）；
- 针对DAS存储融合，应用层逻辑卷与存储LUN之间采用DHT分布式打散映射，以及一致的RAID保护；
- 针对SAN存储融合，应用层逻辑卷与存储LUN之间采用DHT分布式打散映射（新建卷），或者直接映射（利旧并平滑迁移已有

卷），数据可靠性一般由SAN存储自身负责；

➤ 同一应用Volume的直接映射卷可“逐步”平滑迁移到DHT映射卷，实现业务中断。

2.2.3 应用无关的可靠性保障技术

数据中心内的可靠性保障技术

数据中心内的可靠性保障技术主要包括HA（High Availability）冷备份、FT（Fault Tolerance）热备份、轻量级FT。

HA（High Availability）冷备份：数据中心内基于共享存储的冷迁移，在由于软件或硬件原因引发主用VM/PM故障的情况下，触发应用在备用服务器上启动；适用于不要求业务零中断或无状态应用的可靠性保障（见图2-13）。

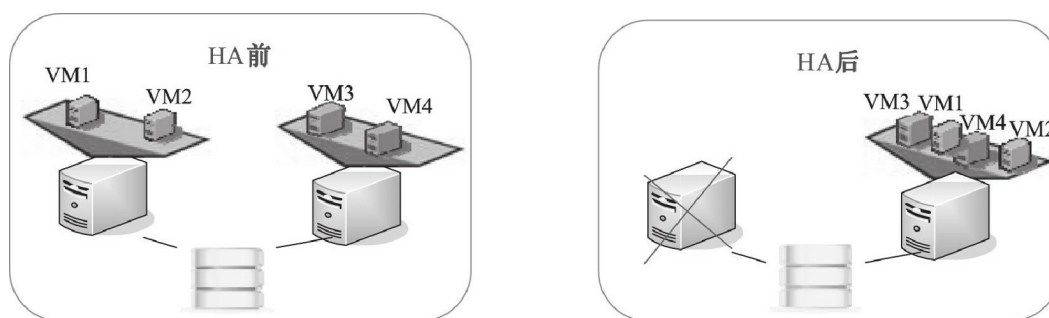


图2-13 冷备份原理

FT（Fault Tolerance）热备份：指令、内存、所有状态数据同步。该方式的优势是状态完全同步，完全保证一致性，且支持SMP。劣势是性能开销大，会带来40%左右的性能降低（见图2-14）。

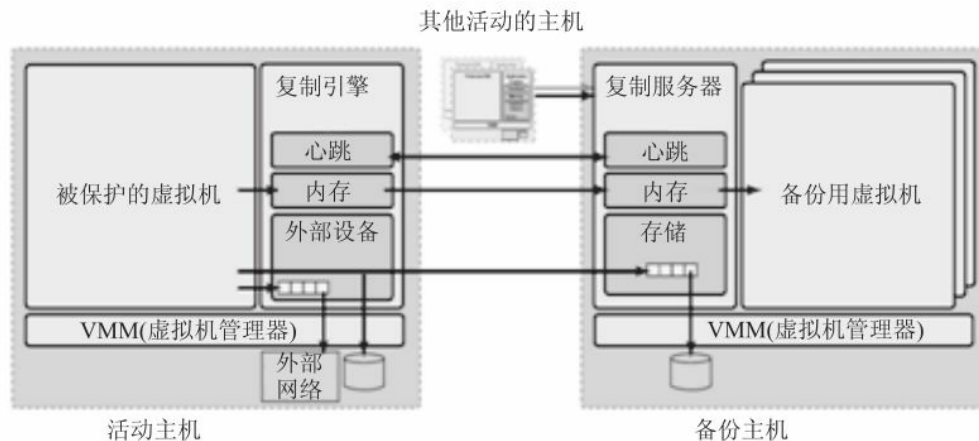


图2-14 热备份原理

轻量级FT：基于I/O同步的FT热备机制。优势是CPU/网络性能损耗10%以内，支持单核和多核。劣势是适合于网络I/O为主服务的场景（见图2-15）。

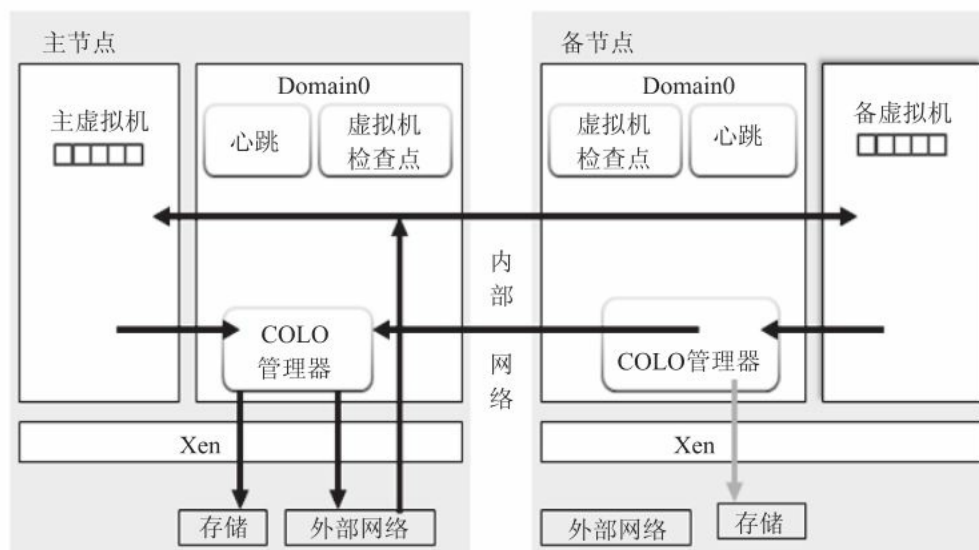


图2-15 轻量级FT原理

（1）跨数据中心的可靠性保障技术

跨数据中心的可靠性保障技术，主要是基于存储虚拟化层I/O复制的同步和异步容灾两种。

基于存储虚拟化层I/O复制的同步容灾，采用生产和容灾中心同城

($<100\text{KM}$) 部署, 时延小于 5ms , DC间带宽充裕, 并且对RPO(恢复点目标)要求较高, 一般RPO接近或者等于0秒。分布式块存储提供更高效率的I/O同步复制效率(见图2-16)。

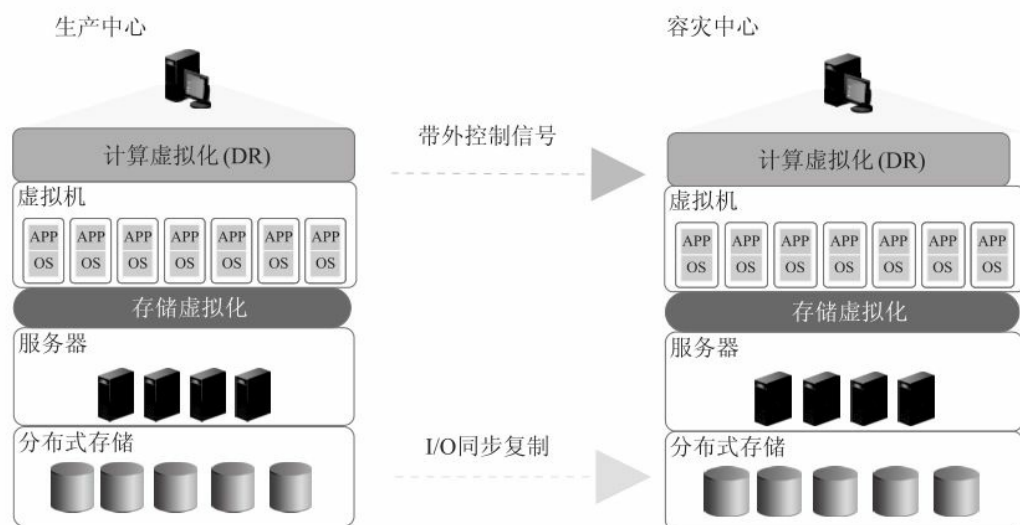


图2-16 基于应用层的容灾复制原理

基于存储虚拟化层I/O复制的异步容灾采用生产和容灾中心异地(大于 100KM)部署, 带宽受限, 时延大于 5ms , 同时对RPO有一定的容忍度, 如RPO大于5分钟。I/O复制及快照对性能的影响趋近于零(见图2-17)。

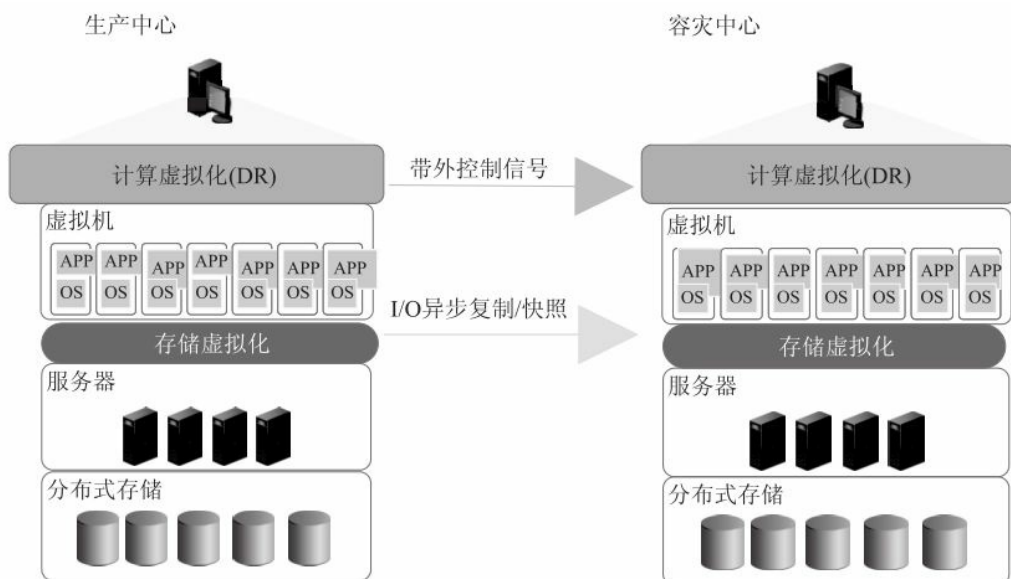


图2-17 基于存储层的容灾复制原理

2.2.4 单VM及多VM的弹性伸缩技术

单VM及多VM的弹性伸缩技术包括基本资源部件级别、虚拟机级别、云系统级别三个层次的伸缩技术。

基本资源部件级别：精细化的Hypervisor资源调度，对指定虚拟机实例的CPU、内存及存储规格进行弹性伸缩，并可对伸缩上下限进行配额限制。

虚拟机级别：指虚拟机集群的自动扩展与收缩，基于CloudWatch机制对集群资源忙闲程度的监控，对业务集群进行集群伸缩与扩展的Auto-Scaling控制。

云系统级别：在内部私有云资源不足的情况下，自动向外部公有云或其他私有云（计算及存储资源池）“租借”及“释放”资源。

上述弹性伸缩机制，使得在大规模共享资源池前提下，流控及因流控引发的业务损失被完全规避（见图2-18）。

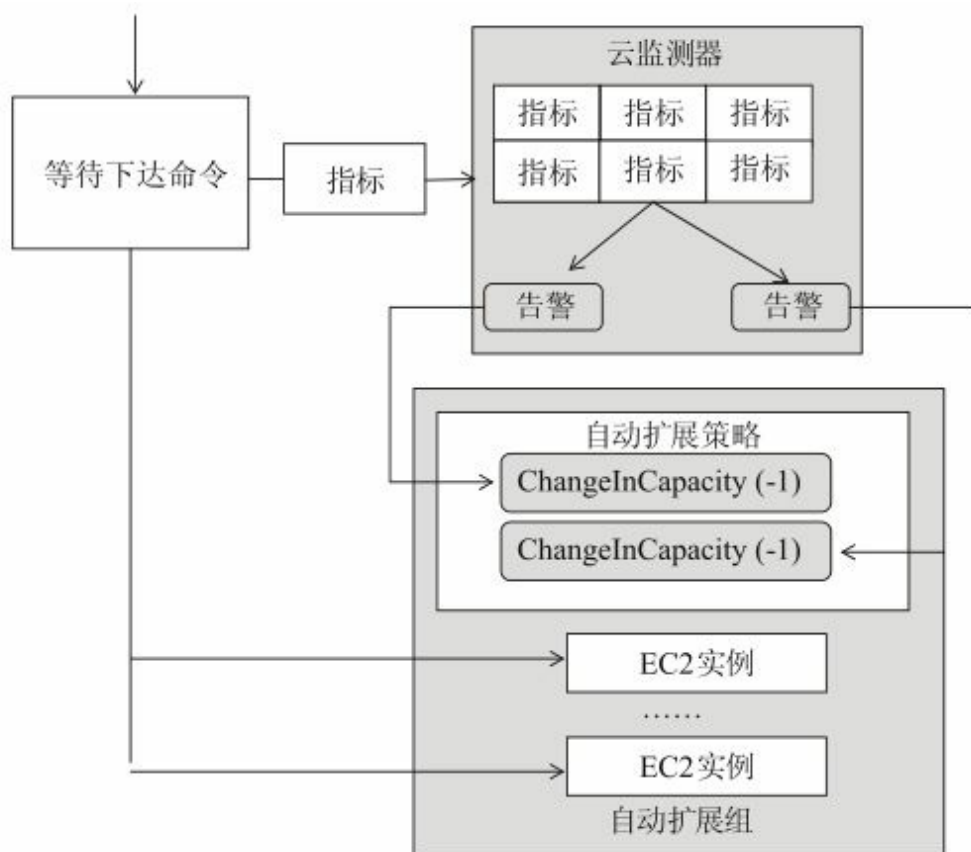


图2-18 弹性伸缩

2.2.5 计算近端I/O性能加速技术

原则上，针对在线处理应用，I/O加速应发生在最靠近计算的位置上，因此作为提高I/O性能的分布式Cache应该运行在计算侧（见图2-19）：

- 远端Cache的I/O效率，高出本地IOPS/MBPS效率1个数量级；
- 通过分布式内存、SSD Cache，实现对内部和外部HDD硬盘介质资源的I/O性能提升2~3倍；
- NVDIMM/NVRAM和SSD Cache保证在全局掉电（或多于2个节点故障情况下）情况下计算近端的写Cache数据无丢失；
- 分布式Cache可提供更大的单VM（单应用）的磁盘并发MBPS，效率可提升3~5倍。

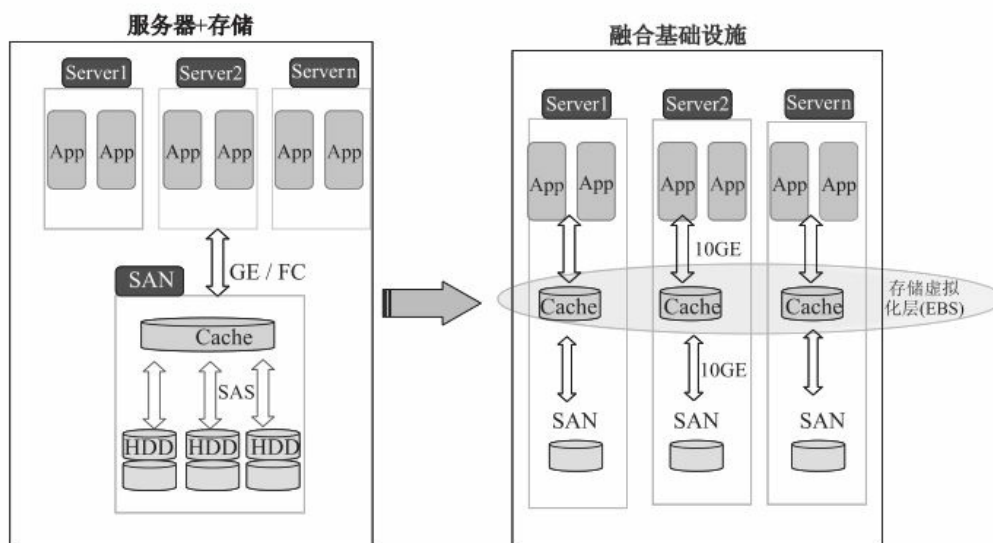


图2-19 存储加载功能

2.2.6 网络虚拟化技术

1. 业务应用驱动的边缘虚拟网络自动化

分散在交换机、防火墙、路由器的L2转发表及L3路由表集中到SDN控制器，使得跨多节点的集中拓扑管控及快速重定义成为可能。基于x86的软件交换机和VxLAN隧道封装的Overlay叠加网可以实现业务驱动且与物理网络彻底解耦的逻辑网络自动化，支持跨数据中心的大二层组网。基于业务模板驱动网络自动化配置，如图2-20所示。

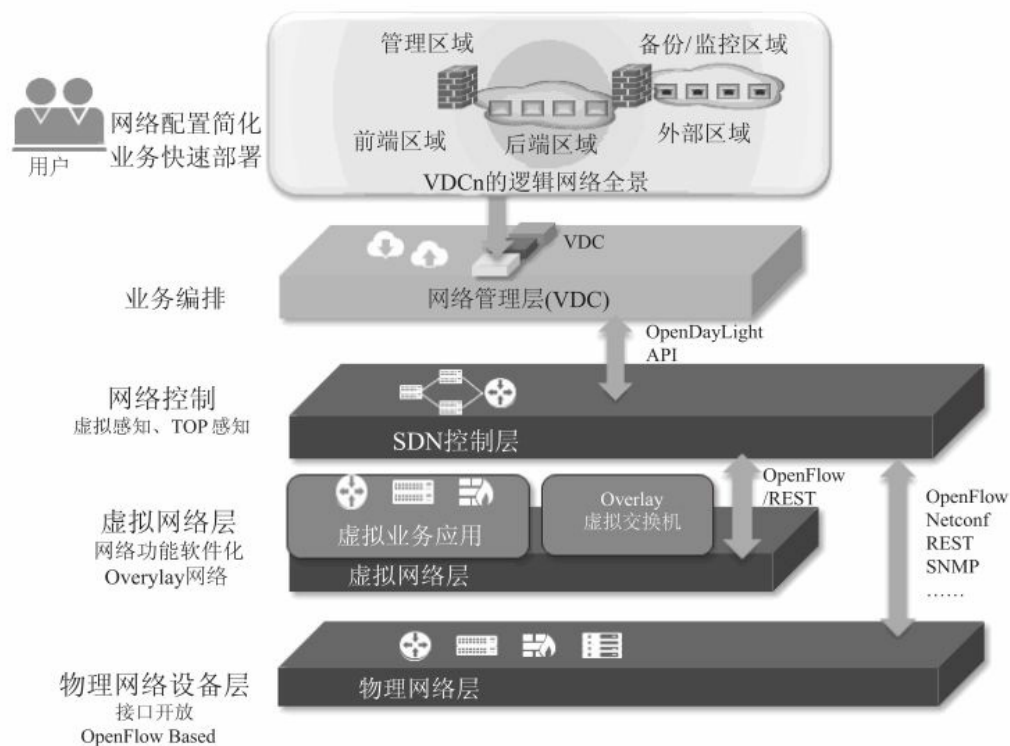


图2-20 网络功能虚拟化层次架构

2. 更强大、更灵活的网络安全智能策略

在云计算早期阶段，一般采用下面的方法进行网络安全的部署，但存在一些不足。

- 公有云、多租户共享子网场景下，静态配置安全组规则仅在目的端进行过滤，无法规避DOS攻击（见图2-21）。

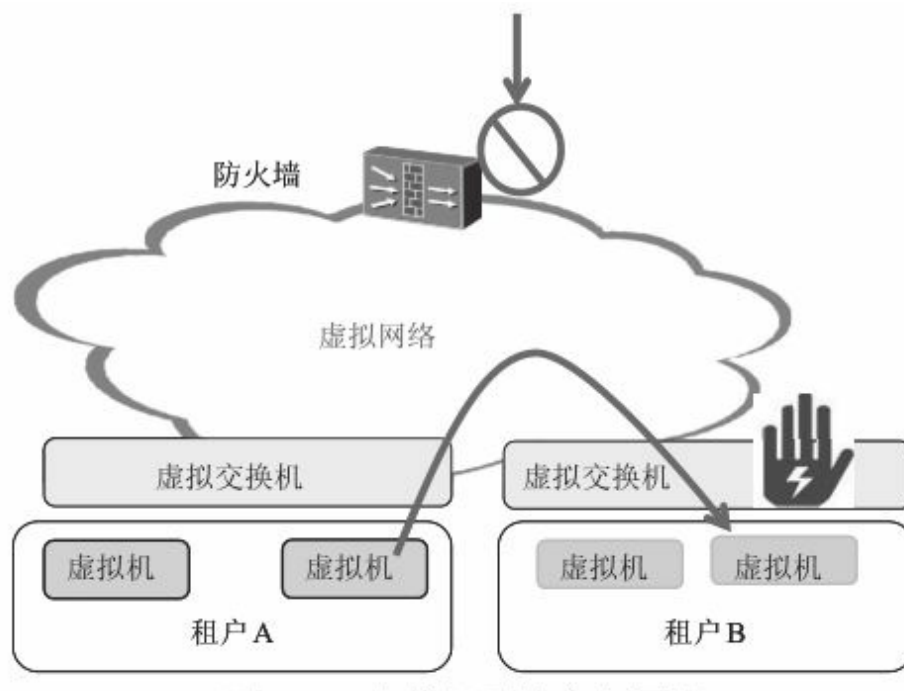


图2-21 虚拟机网络安全原理

➤ 采用外置防火墙方法控制子网间安全，但存在流量迂回（见图2-22）。

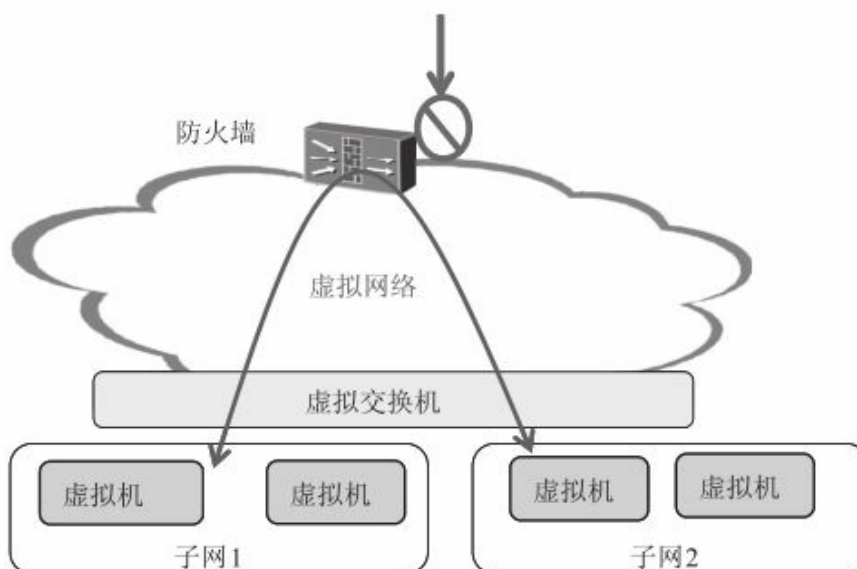


图2-22 采用外置防火墙方法控制子网间安全

为解决云计算早期阶段技术安全隐患，新阶段云计算架构通过软件定义

网络的实施，解决这些问题（见图2-23）：

- 按业务需求统一定义任意目标——源组合的安全策略定义下发到Controller；
- 子网内互访，首包上送Controller，动态下发安全过滤规则，源头扼杀攻击；
- 子网间互访，动态下发快转流表，避免迂回。

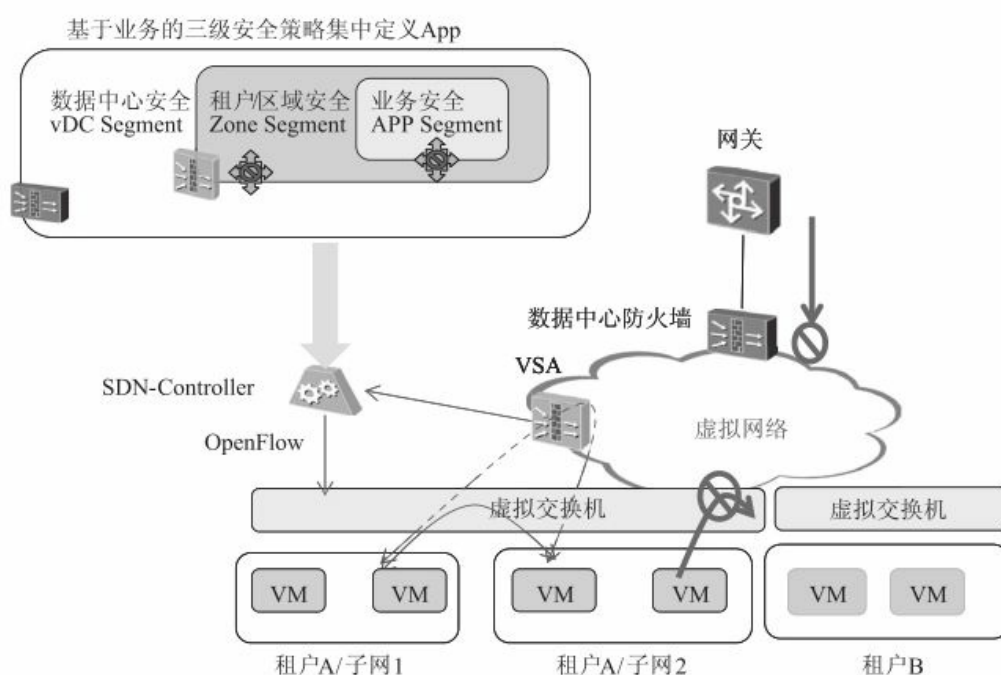


图2-23 通过软件定义网络的实施解决问题

2.2.7 应用管理自动化技术

对于目标架构的基础设施层的管理功能定位，仅仅做好物理和虚拟机资源的调度是远远不够的，其应当涵盖独立于具体业务应用逻辑的普遍适用的弹性基础设施之上的应用的全生命周期管理功能，涵盖从应用模板、应用资源部署、配置变更、业务应用上线运行之后基于应用资源占用监控的动态弹性伸缩、故障自愈以及应用销毁的功能。整个应用的生命周期管理应遵循如图2-24所示的流程。



图2-24 全生命周期管理流程

各部分主要实现如下内容。

（1）图形化的应用模板设计方式：采用基于图形的可嵌套式重用模板；采用拖拽和粘贴拷贝的方式来定义分布式应用模板；使得模板设计简单高效（见图2-25、图2-26）。



图2-25 图形化的应用模板

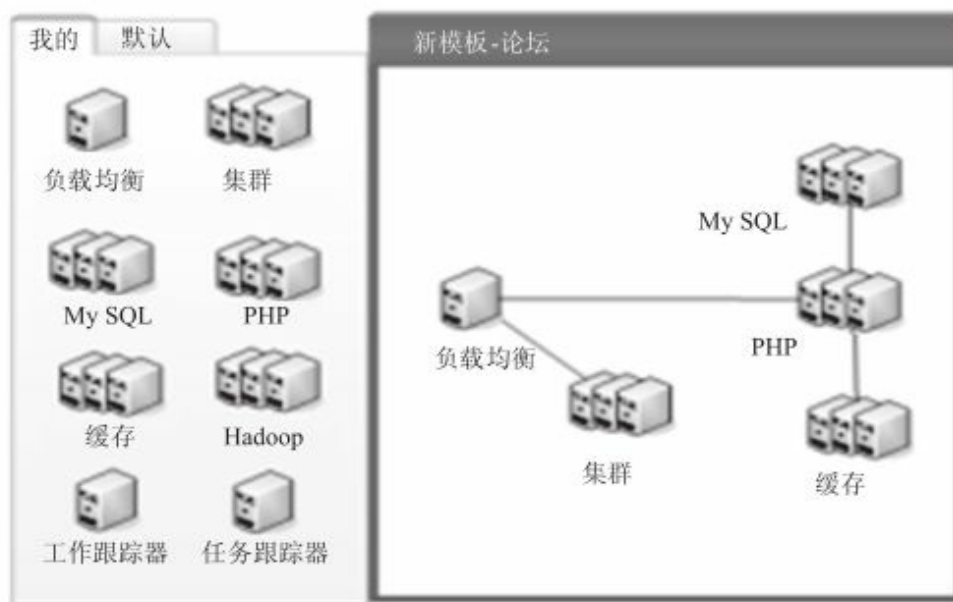


图2-26 图形化的应用模板设计

(2) 提前准备的丰富模板库和自动部署：为物理机、容灾、SDN、LB、防火墙等准备好应用模板；当有应用需求时，系统直接从模板库中选取相应的模板进行自动部署（见图2-27）。

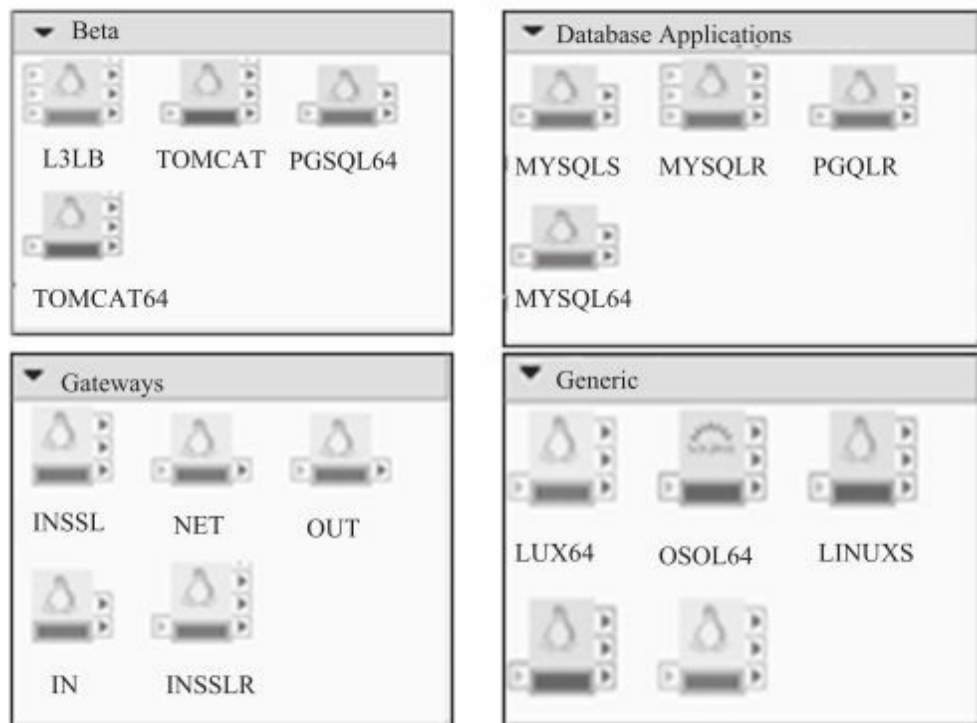


图2-27 从模板库中选取相应的模板

（3）基于SLA的应用监控：面向不同的应用（数据库、HPC、基于LAMP的Web网站等），定义不同的SLA指标集，对这些指标进行监控，采用静态阈值和动态基线相结合的方法进行故障告警和性能预警，使应用监控更自动化和精细化，满足客户业务运行的要求。

（4）基于工作流的应用故障自愈：采用基于工作流的管理方式，通过对应的设计工具来设计用户自定义事件，当监控到应用故障、事件触发和工作流引擎的运转时，系统支持应用的自动修复，达到故障自愈的目的。

2.3 云计算核心架构的竞争力衡量维度

从将云计算技术引入传统数据中心所带来的独特商业价值角度看，重点可以从开源与节流两个方面来衡量云计算的核心竞争力。

节流（**Cost Saving**）方面

在业务系统搭建过程中，云计算和虚拟化使得企业及运营商的烟囱式软件应用可以突破应用边界的束缚，充分共享企业范围内、行业范围内甚至全球范围内公用的“IT资源池”，无需采购和安装实际物理形态的服务器、交换机以及存储硬件，而是依赖于向集中的“IT资源池”动态申请所需的虚拟IT资源（或资源集合），就可以完成相关应用的自动化安装部署，从而达到快速搭建支撑自身核心业务的IT系统与基础平台的目的。这种模式可以减少系统搭建的人力和资源投入，降低系统初始构筑成本。

在业务应用执行过程中，依托节能减排及资源利用率最大化原则，实现必要的智能资源动态调度，以完成既定的业务处理或计算任务，并在特性业务处理或计算任务完成后即时地释放相关IT资源供其他企业、行业进一步动态共享，从而实现IT建设与运维成本的大幅度优化与降低。

另外，针对涉及海量数据处理及科学计算的特殊行业，以往依托于造价昂贵小型机、大型机甚至巨型机、高端存储阵列，或者采用通用处理设备需要数月甚至数年才能完成的复杂计算与分析任务，有可能在云计算数据中心基于通用服务器集群，以更为低廉的成本并花费更短的时间就可以轻松应对。

开源（**Revenue Generation**）方面

➤ 针对公有云数据中心运营商的价值：将SaaS等早在云计算概念出现就已普及的资源服务的概念进一步扩展到IaaS与PaaS层，云计算数据中心运营商可以在IaaS/PaaS上建设自营增值业务服务于云用户，也可引入众多第三方应用运行在IaaS/PaaS云平台之上，实现相比传统数据中心托管服务具备更高附加值的虚拟机、虚拟桌面及虚拟数据中心租赁业务，或者在第三方应用开发/提供商、云运营商（IaaS/PaaS云平台提供者）以及云租户/云用户之间分享丰富的SaaS应用以带来的增值利润。

➤ 针对企业私有云数据中心建设的价值：云计算使得IT基础架构可以对与企业、行业业务紧密绑定的业务软件形成更为高效和敏捷的集成融合，从而大大提升企业IT资源灵活适应并支撑企业核心业务流程与业务模式快速变化的能力，有效地优化企业业务的运作效率。

➤ 云计算的海量数据分析与挖掘能力的价值：使得企业、行业有能力依托其海量存储及并行分析与处理框架的能力，从其企业IT系统所产生的海量的历史数据中提炼并萃取出对其有价值的独特信息与价值，从而为其市场及业务战略的及时优化调整提供智能化决策引擎，从而有效提升企业的竞争力。

基于以上云计算数据中心解决方案商业价值，可以从下面六大架构质量属性指标来衡量云计算数据中心解决方案的竞争力（见图2-28）。

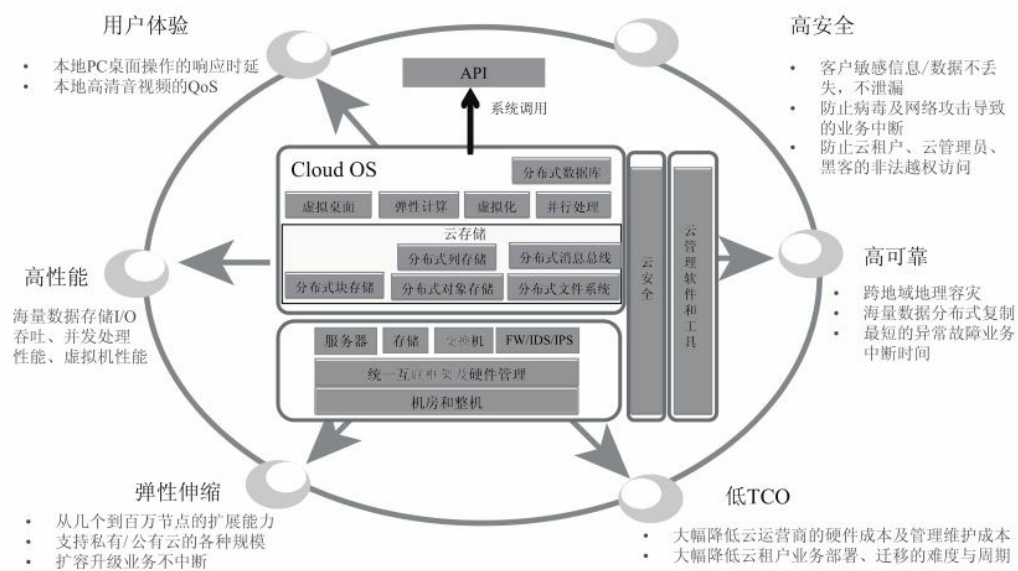


图2-28 云计算架构核心竞争力

2.3.1 低TCO

低TCO能力的构建包括降低和优化云计算数据中心的设备投资成本，以及运维成本两个方面。

设备投资成本的优化与节能主要考虑的架构策略涉及如下策略。

计算资源的成本优化与节省

站在整体数据中心资源集群成本的视角，针对由以太网交换网连接起来的计算密集型服务器构成的计算集群，云计算资源集群初始调度分配以及运行态动态算法的优劣，决定了通过资源占用的削峰错谷可以带来的资源利用率提升，以及相应的成本下降比例。

除调度算法之外，支撑更大规模的集群是实现计算资源CAPEX成本最优化的有效手段，云计算的云资源调度软件通过支持大甚至是超大规模的HA集群（例如：总集群服务器容量达到128服务器/集群），实现对多种大小不同颗粒度的客户服务器集群的容纳，从而降低资源碎片概率，提升资源利用率。

考虑将成本颗粒度从服务器集群细化层面向下细化到单服务器层面，虚拟机VMM引擎在一个服务器范围内的CPU及内存资源调度能否实现跨虚拟机的充分动态共享，则决定了服务器颗粒度内的多虚拟机资源利用率的高低，以及对应的成本竞争力。虚拟化引擎通过支持实时应用调度优化、小包数据中断调度优化，以及内存气泡、内存交换与共享等优化措施来提高服务器级资源利用率。

存储资源的成本优化与节省

在普通云计算数据中心环境下，存储容量一般均在几十TB以上，在满足相同容量及IOPS存储吞吐能力需求的基础之上，需从成本角度出发做出权衡抉择，即采用基于集中RAID控制器带一系列存储磁盘的垂直扩展（Scale-up）模式，还是采用基于全分布式及普通服务器附带硬盘存储的水平扩展（Scale-out）模式。通过引入全分布式存储，有望通过差异化架构规避RAID双控制机头随存储容量与处理能力的上升所带来的成本指数级增长的矛盾，从而实现云存储成本的大幅降低及性价比的提升。

网络资源的成本优化与节省

在可能的情况下，考虑取消独立硬件形态的汇聚网络交换机及防火墙网关设备，在通用x86平台上支持Load Balancer、防火墙等设备，从而有效降低网络资源的成本占用。由于部分云计算虚拟网络特性（如ACL，安全组等）可能大量消耗CPU资源，需要考虑将相关功能卸载到智能网卡上。

维护成本的优化与节省

为实现数据中心大规模计算、存储集群依据多层网络交换设备的维护成本最优化，要求云管理OSS支持最大限度的智能化管理，实现系统在故障状态下对DC内部服务器、网络及存储资源垂直整合的融合架构，一站式交付将大大降低硬件安装维护复杂度。

节能减排等生命周期维护成本的节省

为达到数据中心整个运行服务周期中节能减排效率的不断提升，包括在完成相同工作负荷的前提下更为有效地降低服务器、存储及交换设备自身的耗电量，可采用以下几项关键措施：

- 在云管理层面引入更为优秀的资源调度算法，通过热迁移机制实现将轻载应用尽量合并到数量更少的服务器上，其他服务器则直接下电，从而提升整体资源利用率；
- 在服务器颗粒度内，引入多级节能控制机制，在轻工作负载时自动调整CPU工作于节能状态；
- 在硬件选型方面尽可能选择低功耗CPU以及器件、组件以构筑低成本优势，不断改善服务器单板散热布局；
- 引入分布式电池或者电容，减少由于UPS在空载或轻载情况下的电源效率损失；
- 在数据中心基础设施层引入更为智能的热管理软件及监测手段，并实现充分的冷热风道隔离，以及热耗散的自动补偿，甚至通过直接拉通来实现整体PUE效率最佳。

2.3.2 弹性伸缩

弹性伸缩要求以相同架构，支撑从最少几个计算与存储节点，到最大10万甚至是100万级的计算与存储节点集群规模，且保证数据中心容量扩展过程中的业务连续性 & 业务服务不中断，或中断时延最短。

这里的弹性伸缩扩展能力应该体现在：

- 管理节点的弹性伸缩能力；
- 数据中心资源的弹性伸缩能力；
- 所承载云租户业务的计算集群弹性伸缩能力；

- 承载用户数据信息及系统卷镜像的存储集群的弹性伸缩能力；
- 连接计算与存储集群资源的网络弹性伸缩能力。

为了支持该能力，数据中心交换枢纽需要支持大二层虚拟化网络，采用CLOS无阻塞以太网连接模型；存储资源需要支持全分布式存储架构；计算资源需要支持大集群规模；管理节点需要支持基于共享存储、无状态机制实现的可无级扩展的管理能力。这些要求是支撑该强弹性伸缩能力的根本保障。

2.3.3 高性能

整体云计算的性能，重点体现在以下几个维度。

- 虚拟化云平台上运行普通颗粒度托管应用场景：I/O吞吐性能、CPU调度效率等相比同等处理能力物理机平台的下降比例越小，性能竞争力越强。
- 并发云平台上运行超大规模数据分析与应用处理场景：完成既定任务所耗费的时间越少，或计算、存储资源越少，性能竞争力越强。
- 频繁数据库操作或媒体类存储信息的应用场景：云存储的IOPS吞吐率最大，可共享的存储容量越大，性能竞争力越强。
- 云计算数据中心内依赖于网络总线的分布式B/S、C/S应用（如网站）场景：网络时延越短，性能竞争力越强。
- 批量虚拟机发放、虚拟机系统加载等涉及大流量、大尺寸数据流及文件处理的场景：需要通过包括P2P加载、链接克隆等有效架构手段加以优化，或通过Cache机制减少跨节点性能压力。其依托弹性计算及分布式存储与中间件的并行数据分析引擎，支持批处理及流式海量数据分析与挖掘，从而提升性能。

2.3.4 领先的用户体验

以桌面云应用场景为例，通过局域网络或者远程网络连接的桌面业务体

验，包括基础桌面应用操作、音视频播放、VoIP、高清视频以及3D加速图形处理密集型应用，达到与本地桌面体验持平，并在时延、抖动、带宽占用等方面有优势，这决定了直接面向企业及个人家庭最终用户的云业务的质量保障的评价。

2.3.5 高安全

安全性无疑是云计算技术在数据中心建设部署与扩容中被采纳的首要障碍性因素，尤其是公有云场景，该方面的问题更为突出，因而其也是架构竞争力衡量的关键纬度。对安全质量属性的需求实际上贯穿于云计算架构的自低向上的各个层面，包括：

- 物理层的数据中心安全防护，实现硬件层与软件层关联可信度管理的TPM机制；
- 虚拟化层公共的事件检测、防病毒及安全管理机制；
- 操作系统安全加固，去除无用服务及安全隐患；
- 面向云租户/云用户的云资源管理接入认证与加密管理；
- 面向云资源管理维护者的分权分域及认证鉴权管理，以及面向虚拟私有云的分级资源管理授权能力；
- 面向云租户/云用户的数据传输加密、解密及网络层安全隔离机制；
- 面向云租户/云用户，以及云管理员的数据中心边界安全网关、防火墙或者地址隐藏转换机制（集中硬件或分布式软件）；
- 面向云租户/云用户的云存储持久化数据加密、解密及其密钥管理机制；
- 作为云管理维护及云用户交互界面的Web Portal的应用层安全防护攻击机制；

- 面向云管理者及监管机构的，基于数据中心系统管理与业务日志的安全合规性分析；
- 托管应用的安全网关（如Email、其他Web应用等，可选）。

2.3.6 高可靠

更高级别的可靠性一直被公认为是集约式的数据中心计算模式，相比传统全分布PC计算模式，其可以提供货架式的、具备量化服务水平保障的增值特性。由于云计算技术的引入，使得数据中心系统的动态计算负载可以进一步以与上层应用无关的形态进行跨越硬件服务器边界的调度，同时数据信息也可通过网络在数据中心内不同持久化存储和计算节点内存之间，甚至是跨地理区域的多个不同数据中心之间进行容灾同步，使得运行于云平台之上的应用相比传统方式，可将更多的精力聚焦于核心业务，而将可靠性保障留给云计算数据中心IaaS服务层来提供。

高可靠性的架构属性保障涵盖如下方面。

- 云管理节点自身的可靠性保障机制。
- 承载用户计算负载的计算节点的故障恢复机制：计算节点本地重启故障，以及不可本地重启类的异地恢复类故障发生时，如何在无需维护干预以及应用层特殊处理的前提下，保持业务提供的连续性。
- 云计算数据中心整体网络的可靠性保障机制。
- 云存储数据连续服务与数据防丢失保障机制，如硬盘故障、服务器故障、机柜/机框，乃至整个数据中心意外电源及网络故障的容错与恢复能力。

2.4 云计算解决方案的典型架构组合及落地应用场景

2.4.1 桌面云

基于云计算总体架构下的桌面云解决方案，如图2-29所示。

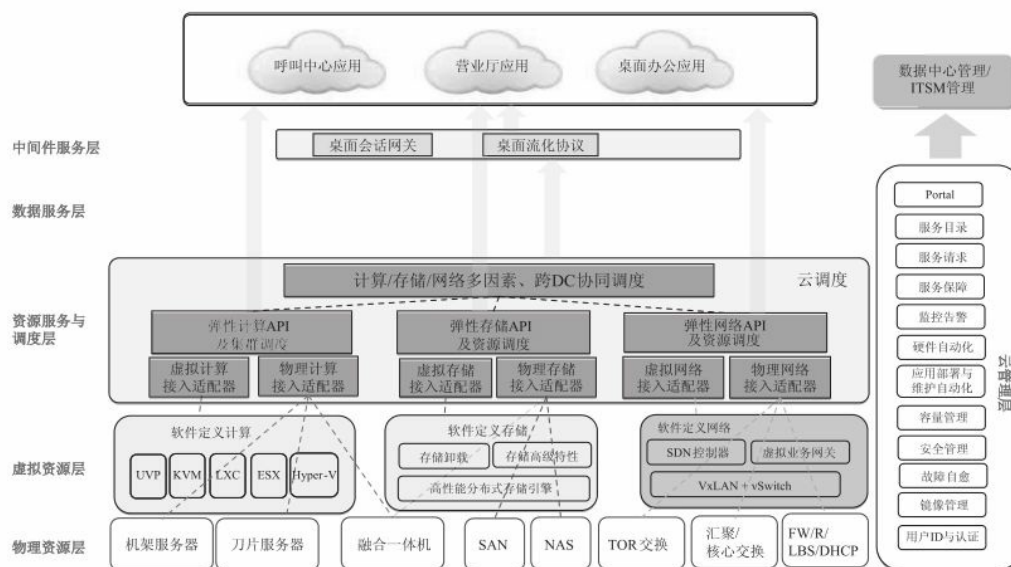


图2-29 桌面云解决方案架构子系统组合

桌面云解决方案主要基于云计算平台的弹性计算、弹性存储、操作运维及业务发放管理系统功能，通过集成桌面云会话控制管理及远程虚拟桌面控制代理等模块，提供针对企业内部应用的呼叫中心桌面云、营业厅终端桌面云、Office办公桌面云解决方案，以及面向公众网用户的VDI出租业务。

其主要业务流程包括：

- 来自企业IT系统或运营商BOSS系统的桌面云发放命令，通过与云计算云管理平台之间的SOAP、RESTFUL API接口，交互包含标准桌面配置规格定义的VDI业务发放/撤消封状式命令；
- 云管理平台的BSS系统对VDI业务发放命令进行解析，将该命令分解为指向“桌面会话网关”的VDI账户发放命令，以及指向“弹性计算API及集群调度”的VDI虚拟机实例发放命令；
- “桌面会话网关”接受来自云管理平台的命令创建桌面账户；
- “弹性计算”服务部分接受来自云管理平台BSS部分通过的EC2兼容接口创建符合原始发放需求规格、包含虚拟桌面服务器端代理的

虚拟机镜像，桌面服务端的EC2 IP地址还将反馈给“桌面会话网关”系统；

➤ “桌面会话网关”接收来自VDI瘦终端的HTTP/HTTPS桌面会话登录请求，通过云计算API与AAA服务器交互，或者与企业LDAP/AD服务目录交互进行用户身份鉴权，随后进一步通过EC2 API通知弹性计算服务启动虚拟机的运行实例；

➤ “弹性计算”服务通过与“弹性存储”通过SOAP消息交互，完成指定业务发放命令中指定的虚拟存储卷的挂载，其中包括为该虚拟分配的系统卷及数据卷，虚拟机从其系统卷引导启动，完成系统的初始化启动；

➤ 用户认证通过后，将用户所用VDI瘦终端通过远程桌面协议与数据中心服务器相连接，建立桌面会话；

➤ 来自瘦终端的客户操作通过“桌面流化协议”与其后端服务的虚拟机实现交互，完成实际的操作动作；

➤ 构成桌面云解决方案的弹性计算管理服务器、块存储及对象存储设备，VDI虚拟机，桌面云会话控制网关，数据中心各层交换机（接入/汇聚/骨干），防火墙，桌面云特有的负载均衡及Cache加速设备（LBS）相关的监控、告警、性能、配置、拓扑以及安全管理等，均通过符合SOA原则的Web OM对象化API接受云管理子系统的端到端管理；

➤ “弹性计算”的智能调度算法，可以为有效提升桌面云VDI VM在整体服务器集群内的资源利用效率，减少负载不均衡导致的资源浪费并提升节能减排效率，以及实现桌面云工作负载与其他非桌面类工作负载的动态调度能力提供支撑；

➤ 由于“弹性计算”内所有服务器及虚拟机共享相同的“弹性存储”实例使得支持业界最大规模的支持虚拟机HA及在线热迁移的集群成为可能，在集群内的服务器故障可以快速恢复，从而实现软硬件故障导致的VDI服务故障影响最小化；

➤ 由于“弹性存储”所提供的块存储跨服务器、跨机柜的数据可靠性冗余机制（2份或3份拷贝）可以为桌面云功能提供超越PC本地存储的数据可靠性保障，同时基于对象存储的快照机制以及异地容灾机制，使得桌面云数据在更大灾难发生时也有机会还原为最近一次快照时刻的存储数据内容；

➤ 针对桌面办公类应用，采用软件预安装模式，除基本客户机操作系统外，进一步增加终端安全管理、Office系列、UC通信、Email、CRM、ERP等预装软件，并可根据不同目标市场，制作不同的虚拟机模板，可以在云管理平台中指定不同虚拟机、存储及网络带宽规格，甚至不同的计费规则（针对公有云桌面出租的场景）。

2.4.2 存储云

基于云计算总体架构下的存储云解决方案，如图2-30所示。

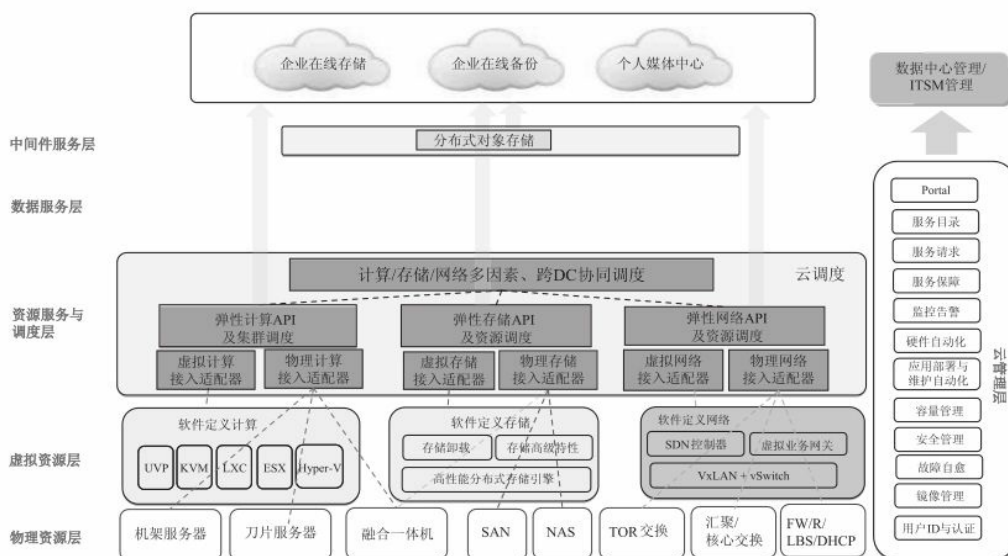


图2-30 存储云解决方案架构子系统组合

云存储解决方案依托云计算平台的弹性存储、分布式对象存储，以及操作运维及业务发放管理系统等功能，通过集成第三方的企业在线备份软件、个人网盘、个人媒体上载及共享类软件，允许云存储运营商提供面向个人消费者用户的廉价/高性能网络存储服务，以及面向企业用户的在线备份及恢复类服务。

在云存储平台与企业备份/恢复类应用软件以及个人网盘、个人媒体上载/共享类软件绑定部署销售的场景下，与云存储应用用户的Portal、UI交互界面及其与应用相关的核心业务功能（如权限管理、断点续传等）由第三方合作的应用软件支撑，同时从服务器端或直接从客户端调用“弹性存储”服务的对象存储或分布式文件系统API（OBS/POSIX），实现对云存储用户的高吞吐量、超大容量存储内容的读写及其元数据管理。该模式下，用户的计费主要由第三方软件负责。

运营商的云存储平台以IaaS形式提供与第三方合作的企业备份/恢复类应用软件，以及个人网盘、个人媒体上载/共享类软件的后端支撑，此时“弹性存储”提供对第三方云存储应用软件的多租户隔离以及存储空间和IOPS/MBPS访问流量的精确计量，以便为云存储服务商与第三方增值服务提供商之间的计费结算与商业分成提供支撑。

“弹性存储”提供以通用服务器及其硬盘为基础的全分布式平台，具备水平无级扩展、超大容量等特点，并通过瘦分配、跨用户的重复数据删除、数据压缩等大幅降低云存储的设备及运维成本，实现超高性价比存储方案，提升云存储类业务的利润空间。

“弹性存储”所提供的块存储跨服务器、跨机柜的数据可靠性冗余机制（2份或3份拷贝）可以为存储云功能提供超越PC本地存储的数据可靠性保障，同时基于对象存储的快照机制和异地容灾机制，使得存储云数据在更大灾难发生时也有机会还原为最近一次快照时刻的存储数据内容。

2.4.3 IDC托管云

基于云计算总体架构下的IDC托管云解决方案，如图2-31所示。

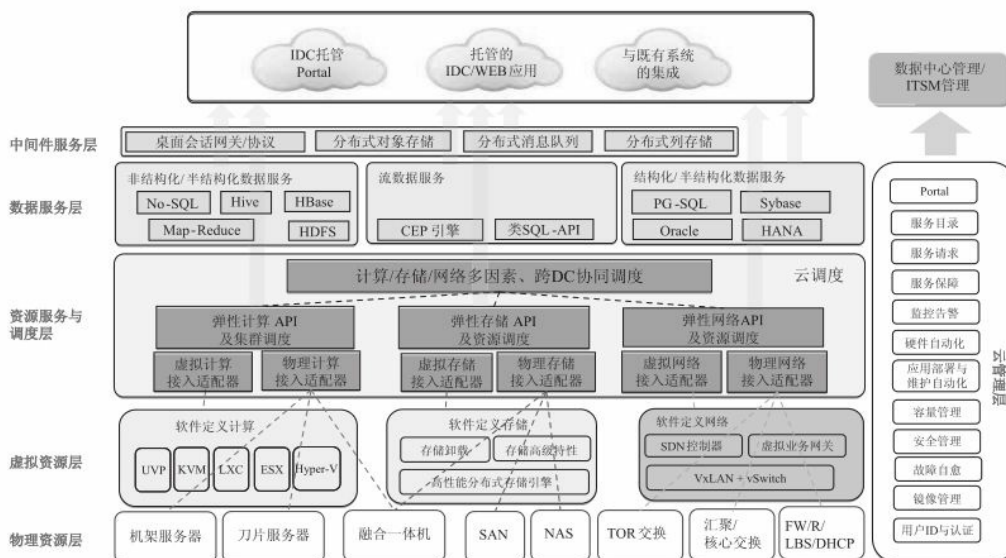


图2-31 IDC托管云解决方案架构子系统组合

IDC托管云解决方案依托云计算平台的弹性计算集群、弹性存储集群、分布式结构化存储服务以及分布式消息队列服务，为IDC（Internet数据中心）运营商提供ISP/ICP多租户的计算与存储资源的托管服务。

业务应用与IT子系统运行于IDC托管云中，隶属于不同的第三方ISP/ICP以及企业。IDC托管云依托于云平台的自动化、虚拟化基础能力，实现多租户的分权分域的安全隔离及资源共享，相比传统的物理服务器独占式的IDC解决方案，其可提供高出3~5倍的IDC出租资源利用效率，从而有效提升IDC托管类业务的利润率。

云计算中云管理平台的BSS子系统为IDC业务的运营发放、资费定价、后计费、实时付费（按资源规格、按时长、按流量，以及上述各类维度的综合）提供了强大的后台支撑。如果用户（运营商）已有BSS系统，可通过云计算API（EC2/S3等的兼容API）与用户已有的后台BSS系统进行对接。

除虚拟化计算集群资源（含虚拟CPU与内存资源，以及挂载于该虚拟机实例之下的系统卷及数据卷块存储资源）之外，云计算平台还提供独立的分布式对象结构化存储，分布式消息队列等超越单机物理处理能力范畴的分布式中间件服务（针对运行于云平台之上的软件），以及远程接入的服务能力“虚拟桌面”（针对云平台业务的直接消费者）。

云计算平台对IDC托管的业务应用的管理是通过业务应用底层的操作系

统来完成的。这些操作系统一般为x86架构，如Windows、Linux以及Unix。IDC托管的业务应用也可以与操作系统一起打包作为一体化的虚拟机镜像使用。只要IDC托管应用本身的颗粒度不超过一个物理服务器的场景，则不存在软件兼容性问题，但需要IDC托管应用的软件管理系统实现与云计算平台API的集成，实现软件安装部署及监控维护从硬件平台到云平台的迁移，并可能依赖“运营维管”子系统的自动化应用部署引擎，实现跨越多个应用虚拟机镜像的复杂拓扑连接的默认模板化自动部署，从而有效提升大型分布软件的部署效率，这是目前云计算IDC托管的主流形态。

针对分布式对象存储、分布式消息队列、分布式列存储数据库等场景，则需要IDC托管应用针对其业务层API进行适配改造，相对难度较高，IDC托管应用一般是云计算平台生态战略联盟内的云应用合作伙伴。

2.4.4 企业私有云

基于云计算总体架构下的企业私有云解决方案，如图2-32所示。

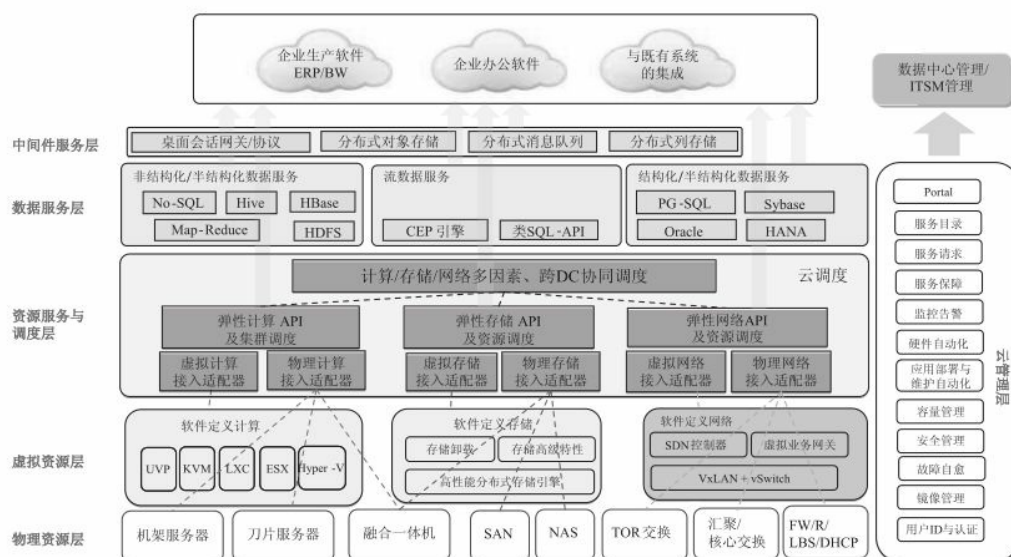


图2-32 企业私有云解决方案架构子系统组合

伴随IT与网络技术的飞速发展，IT信息系统对于企业运作效率、核心竞争力，以及企业透明化治理正在起着越来越重要和无可替代的作用，而企业信息集中化、企业核心信息资产与商业逻辑的规模越来越庞大，跨不同厂家IT软硬件产品的集成复杂度不断增加。企业IT系统的架构正在从传统的与特定厂家硬件平台及管理系统绑定的客户端/服务器（B/S、

C/S)架构向更为集中化的统一整合平台架构的方向演进。云计算平台与企业IT应用层软件的结合，尤其是基于虚拟机的弹性计算服务、虚拟网络服务、虚拟桌面服务、分布式块存储、对象存储服务、文件系统服务以及与之配套的自动化运维管控的能力，使得企业IT系统可以更高效地支撑企业核心业务的敏捷运作，大幅提升IT及机房基础设施利用效率并实现节能减排。

在保障业务运行效率与性能不下降的前提下（计算、存储资源配额，业务访问时延等），通过将原有直接运行在x86服务器硬件平台之上的企业IT软件迁移到虚拟化平台，将企业IT软件相关的存储数据（数据库/文件格式）迁移到分布式块存储或者传统IP-SAN存储，可以充分利用弹性计算平台的跨服务器边界的资源分配与热迁移能力，实现多个相对独立的IT软件应用在虚拟机资源池内动态共享，以及削峰错谷的负载均衡调度，并实现不同应用的分级QoS（硬件资源下限）策略保障，实现IT资源利用效率从平均20%~30%到60%~70%的提升。同时在系统轻载的情况下，通过将轻载虚拟机迁移到少数物理服务器，可实现更多空闲服务器硬件的自动休眠，来最大限度地提升数据中心及IT资源池的节能减排效率。

借助云计算平台的虚拟桌面（即桌面云）能力，可以实现企业员工PC办公的计算与存储能力向数据中心的集中化迁移，实现核心信息资产与用户接入访问终端的解耦和剥离。虚拟桌面具有绿色、节能、安全隔离及移动接入能力方面的优势。除了对办公PC的改造之外，虚拟桌面也是最终企业员工接入到后端IT应用业务的必由界面和通道。

借助面向大型分布式应用程序的云计算自动化、模板化部署，通过故障自动修复管理能力，运行态自动伸缩管理工具，弹性计算、虚拟网络、虚拟桌面与企业IT管理系统（含可选的ITIL子系统）的无缝集成，可以实现IT应用程序与底层IT硬件与网络基础设施的彻底解耦，利用标准化的虚拟应用部署模板（描述格式如OVF）大幅度（70%）缩短IT软件应用的上线部署效率，以及降低业务在线运营的容量规划与故障维护的复杂度，有效提升IT服务支持企业核心业务的SLA水平和效率，从而促进企业生产率的同步提升。

云计算的分布式对象存储、半结构化存储（列存储数据库）以及消息队列能力，对于企业私有云来说，是可选的高层云平台能力。其适用于企业定制开发新型应用，比如：企业/行业搜索引擎，基于企业IT系统海量日志或统计类数据仓库的商业智能挖掘与分析，以便指导企业的业务

规划策略的调整优化等以大数据集作为输入和输出的软件，是性价比最优的选择。但这部分云平台能力在企业私有云中一般无法适用于面向实时在线事务及交易类的应用形态。原因是这些云平台的API与单机通用操作系统（Windows、Linux、Unix等）下的文件系统、进程间通信以及数据库访问API都是不兼容的，而业界大多数企业IT应用软件、商业操作系统以及数据库（如Oracle）软件是运行在通用操作系统之上的。

2.4.5 大数据分析云

基于云计算总体架构下的大数据分析云解决方案，如图2-33所示。

大数据分析云解决方案为海量静态数据批处理以及大流量动态流数据处理为关键特征的企业及行业应用场景提供支撑，通过自动化提取与归纳价值信息实现业务增值。大数据分析云由云计算的并行数据分析与挖掘平台所支撑，可充分利用云计算底层能力创造最大价值。

在海量静态数据批处理的场景下，大数据分析平台需要充分分析经过相当长一段时间积累的、存储容量庞大的历史数据（如话单、日志、话统信息等）。大数据分析平台的并行数据处理引擎进一步依赖于弹性计算集群、弹性存储服务、分布式结构化存储服务以及分布式消息队列服务，为诸如互联网电子商务网站用户、电信运营商的BSS/OSS系统、视频娱乐类网站、搜索类网站等提供服务。大数据分析平台所提供的服务类型包括：信息库精细化搜索、用户消费行为日志分析、系统运行日志分析以及集中监控信令信息的智能分析和挖掘。这些大数据分析服务为精确定位广告推送、网络运维优化、基于用户消费趋势分析的销售策略等商业运营提供策略决策性支撑。

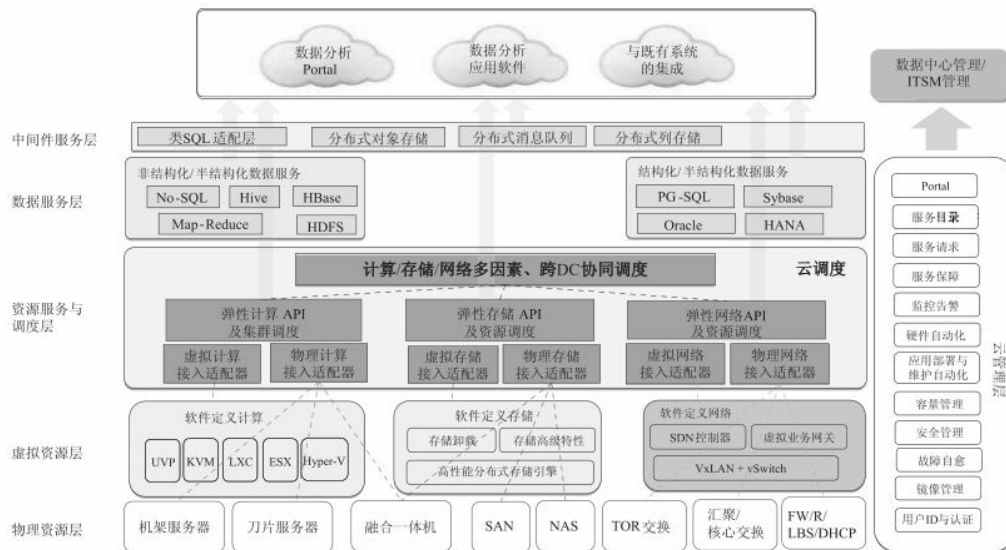


图2-33 大数据分析云解决方案架构子系统组合

在流量动态数据流处理的场景下，其关键特征在于对来自于数量庞大的信息源所产生的动态事件与动态数据（比如来自电信网络实时检测的信令信息、来自于众多车辆GPS的定位信息、来自物联网终端实时采集的信息等），在一个相对较短的时间窗口内，进行动态数据的流水线方式自动关联分析与处理，并给出及时、准确和智能的执行策略决策，为特定业务目标服务（如大规模智能交通云、物流云的构建）。与上节数据批处理在分割、合并与混排等中间步骤所涉及的大量持久化存储I/O交互方面的特征相比，其最大差异处在于，数据流处理过程更讲究处理的及时性与敏捷控制能力，因此处理过程主要在内存中完成。流处理与批处理可以统一在相同的框架引擎之下。

为便于广大第三方应用开发编程人员以及云计算平台生态系统的合作伙伴充分独立于海量数据批处理以及流处理业务的内部实现架构细节，可在并行数据分析引擎与并发应用之间设置SQL/类SQL适配与翻译层，提供开发人员所熟知的SQL或类SQL规范语言进行海量数据的操作。

2.4.6 数据库云

基于云计算总体架构下的数据库云解决方案，如图2-34所示。

数据库云主要指基于云平台构建的结构化/半结构化数据库处理系统。

数据库云可以基于虚拟化平台，也可以基于物理平台直接构建。

在性能要求高的场景下，一般基于物理平台构建，系统无需弹性计算部分（图中虚线表现）；而在要求容量很大、应用用户很多的情况下，则可采用基于虚拟化平台构建的形式。

数据库云一般以数据库一体机的形态出现，会在以下方面做一些增强。

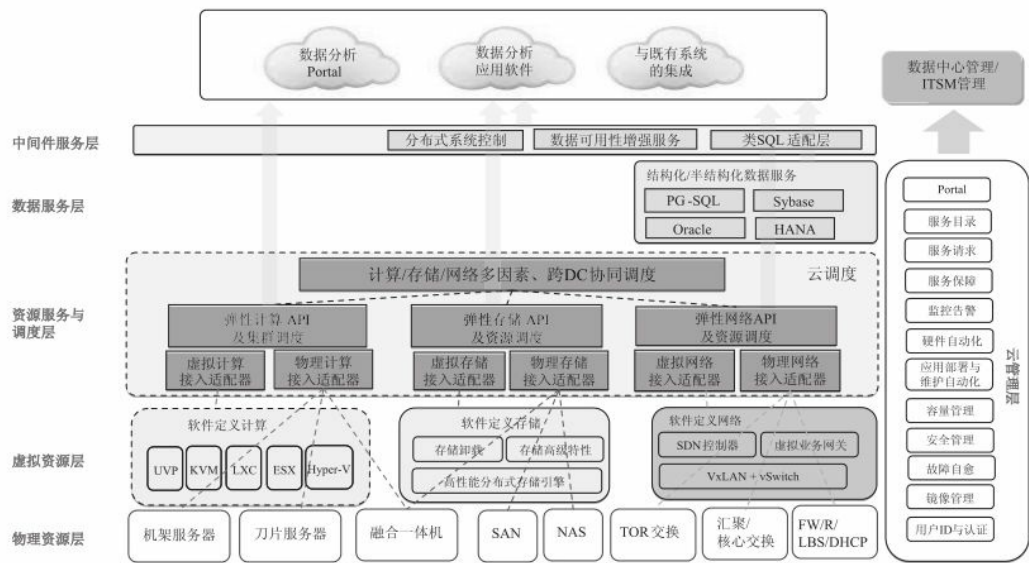


图2-34 数据库云解决方案架构子系统组合

➤ 数据库加速：为取得更好的数据库性能，会在硬件层、弹性存储层做垂直层面的深入调优，例如采用读写更快的SSD卡，采用面向数据库独特的读写算法。

➤ 数据库加固：为保证数据库数据不丢失、不损坏，会在中间件服务层增加数据库的备份/恢复、容灾、定期校验等服务，提高数据的可用性。

2.4.7 媒体云

基于云计算总体架构下的媒体云解决方案，如图2-35所示。

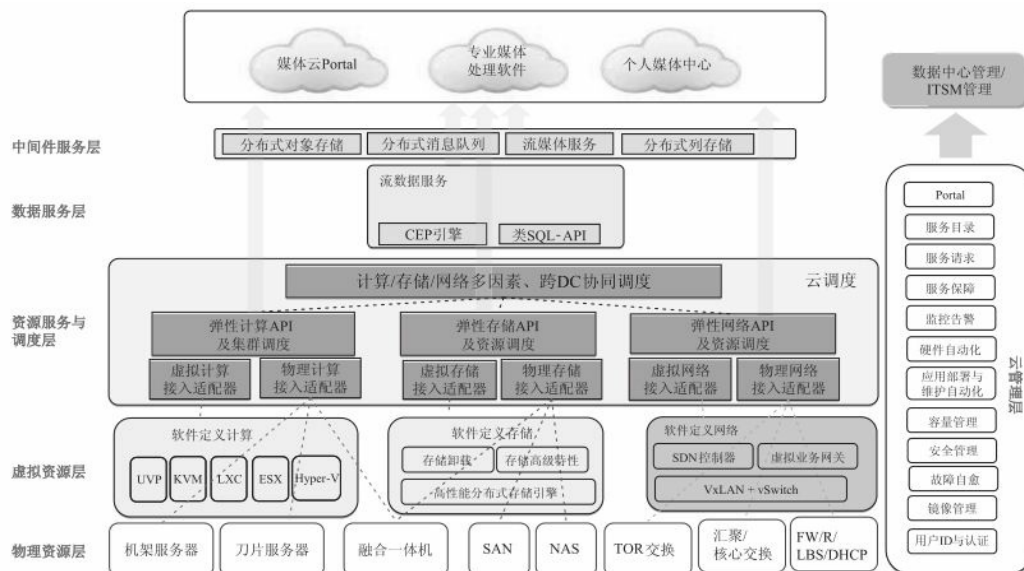


图2-35 媒体云解决方案架构子系统组合

媒体云解决方案依托云计算平台的弹性计算集群、弹性存储集群、分布式结构化存储服务以及分布式消息队列服务，为广电系统电视台企业提供高效率的媒体存储、编辑及缓存加速的计算与存储资源托管服务。

媒体云采用云平台的关键驱动力，来源于面向公共媒体传播的广播电视行业（包括国家及地方电视台、广播电台）从模拟传统卡带式存储向全面的数字媒体化方向的发展演进。在此过程中，数字化内容与云数据信息需进行大规模集约化存储、处理以及分发。这些需求包括媒体的采集、编辑、播放控制、音视频媒体转码、管理等各方面的软件处理需求。目前国内外业界（如中科大洋、索贝等）已纷纷推出针对媒体采集编播的专业化软件，这些软件重点聚焦于解决媒体内容资源的应用层的业务处理，但在对底层的服务器及存储硬件基础能力提升方面仍然缺乏积累，如增强服务器存储硬件的使用效率，如何以更低的成本提供更大的存储空间、处理能力和I/O吞吐带宽，可靠性容灾能力，以及按照业务需求分配弹性可伸缩的资源池等。只有上述专业化的媒体管理软件与云平台能力紧密结合，才能实现媒体数据中心基础设施硬件资源利用率的提升，实现节能减排，获得最优的较硬件整体性价比。

云计算平台对于媒体云数据中心的核心价值观在于以下几点。

- **海量存储能力：**来源于各种片源的音视频媒体内容信息本身的持久化存储，由于数据量庞大、存储周期长（3个月），采用传统

的Scale-up模式的RAID控制方式存储在性价比方面已越来越难以满足媒体云规模化运营的需求。同时，媒体云的多项业务，尤其是视频/音频媒体的编辑制作以及在线播放，都对存储与计算资源之间的高I/O吞吐率（IOPS/MBPS）提出了更高的要求。满足这些需求依赖于引入支持无级水平扩展的Scale-out存储。

➤ 跨业务共享的计算、存储资源共享与均衡：针对媒体云应用层软件的不同业务类别（采、编、播、存、管等）在同一时间段的资源占用情况差别巨大的特点，通过引入云平台，采用统一的资源池平台支撑资源的自动伸缩、动态错峰削谷与负载均衡，通过SOA Web Service/REST接口与应用层软件交互实现自动化的资源管理能力，平均资源占用率可以从20%提升到70%以上。

➤ 并行计算与海量处理能力：针对媒体云普遍所需的不同的视频格式之间的动态转换（TS、H.264、MP4、AVI、RMVB等）需求，由于媒体文件尺寸庞大，计算能力需求密度高，通过最大限度提高编解码的并行度，可充分利用可获得计算资源对大媒体文件分而治之，有效缩短编码处理所需时间，以资源换取时间，大幅度提升业务处理效率。其他诸如广告推送等增值业务，也可依赖于并行数据分析与处理平台实现基于用户消费行为历史数据分析的智能化与自动化的广告策略制定与发布。

➤ 分布式缓存与加速：为缓解广大互联网用户点播热点多媒体内容引发巨量带宽需求与有限互联网广域连接带宽的矛盾，需要在媒体云数据中心对媒体内容进行分片，并自动识别用户访问热点。热点内容被自动推送到分布式网络节点缓存，供用户就近访问，从而缓解集中访问的带宽压力。缓存的内容与云数据中心的源内容会定期同步，以保证用户看到的内容为最新内容。

2.4.8 电信NFV云

基于云计算总体架构下的电信NFV云解决方案，如图2-36所示。

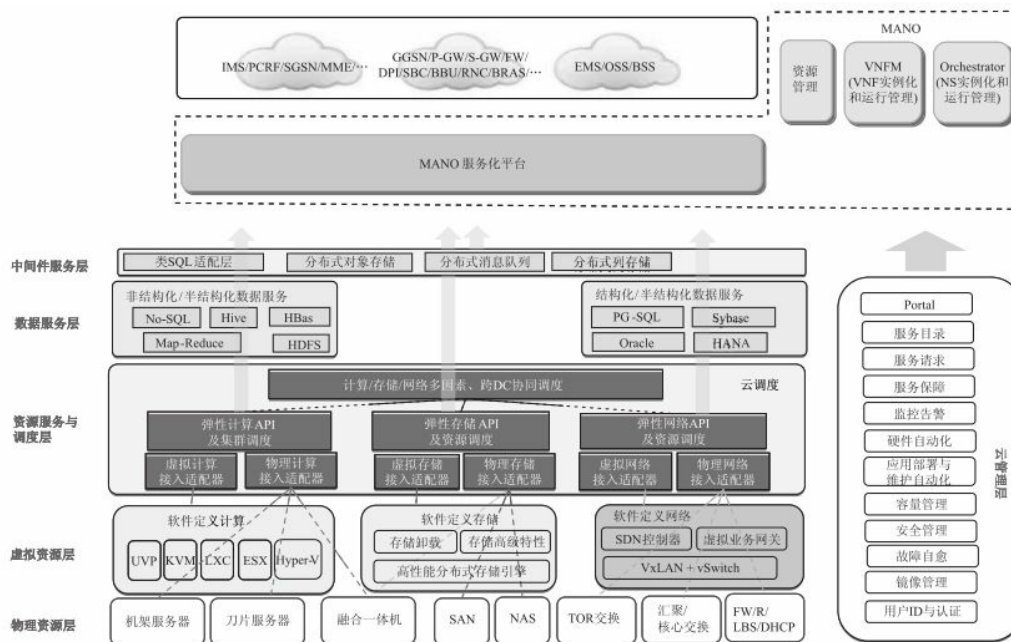


图2-36 电信NFV云解决方案架构子系统组合

NFV（Network Function Virtualization网络功能虚拟化）旨在通过研究标准IT虚拟化技术，使得电信网络设备的功能能够以软件方式运行在符合行业标准的大容量通用的服务器、交换机和存储设备中去。这里的软件可以根据需要在网络中不同位置的硬件上安装和卸载，不需要安装新的硬件设备。简单地说，NFV就是在平台层引入云计算平台，实现电信网元纯软件化和硬件设备解耦。

电信NFV云通过引入云计算平台，主要要解决运营商如下几大问题。

提高硬件投资收益比

简单来说，采用通用硬件一方面可以大大降低成本，另一方面开通新的业务时也不需要更换硬件，只要升级相应的软件即可。

提高业务部署的灵活性

由于NFV使得软件和硬件解耦，也就是说运营商的业务可以灵活部署在不同机框和不同机房，在不同地域的硬件上部署，那么相比传统业务部署来说自然是极大地提高了灵活性。

快速部署业务

软硬件解耦带来的最大好处就是设备商可以专注于纯软件层面的研发，在已有的软硬件部署下，新的业务研发周期会大大缩短，运营商也能尝到快速部署业务的甜头。

自动扩容和节能减排

由于虚拟化技术的支撑，网络智能调度资源的能力大幅提升。在业务压力增加时网络智能调度系统可以通过自动增加网络资源来缓解业务压力；在业务闲暇时可以通过自动地删减网络资源实现节能减排。

降低了设备商的准入门槛

硬件设备通用了，软件接口也通用了，毫无疑问，设备商的准入门槛也降低了，这给电信服务引入了更多的竞争，这些对于运营商来说，都是最想看到的。

NFV希望电信网元纯软件化，同时又需要提供电信级5个9的高可靠性服务，因此对云计算平台需要有更高的要求。

云计算平台对于电信NFV云的核心价值在于：

- 电信应用由于可靠性要求，对应用的虚拟机之间的亲和性关系有着不同的约束，云计算平台支持电信应用的各种亲和性调度需求。
- 电信级服务意味着5个9的高可靠性，对于服务中断时间有着严苛的要求，通过硬件故障快速检测和故障快速通知技术，电信应用能够快速感知故障，并及时进行自愈机制的启动，减少业务中断时间。
- 作为NFV基础设施，提供与应用无关的高可靠性特性：HA（High Availability）特性主要在服务器出故障时提供虚拟机冷备机制，轻量级FT（Fault Tolerance）为面向网络I/O的应用提供热备机制以及跨数据中心的虚拟机容灾机制。
- 针对NFV进行云计算转发面的优化，使得VM到VM之间的转发性能可提升到可以满足电信网元的要求。

➤ 云计算平台利用存储虚拟化技术把所有服务器本地HDD、SAN、NAS组合起来构建超大规模存储资源池，利用服务器的内存/Flash/SSD构建分布式高性能大容量近端cache网格，服务器机头可同步横向扩展，加速I/O读写能力。同时，云存储资源池可提供基于SLA的分级存储服务。

➤ 云计算的自动化和模板化（TOSCA、HOT、CloudFormation、OVF等）特性可大幅度提高电信应用和IT软件应用的上线效率。云计算的故障自动修复和自动伸缩管理能力可大幅度降低业务在线运营与故障维护的复杂度。

➤ 电信NFV是大规模复杂系统，从电信应用到IT的各种应用，包罗万象，服务海量用户，在线生成海量数据。云计算的结构化数据库、分布式对象存储、半结构化存储（列存储数据库）以及消息队列等中间件能力，为电信NFV带来广泛的运营基础能力，满足日益增长的新老应用的基础能力需求，加快新业务和新运营应用的上线能力。

第3章 云计算相关的开源软件

3.1 云计算领域开源软件概览

云计算是一个很大的系统，它的设计实现涉及硬件的实现，涉及虚拟化内核的选择，涉及各种计算/存储/网络虚拟化技术的选择，涉及云资源的申请和管理；不同的公司可能采用不同的方法来实现。然而，云计算的理念就是要提供像电一样的公共产品，那它必然就涉及标准和开放问题，否则各系统无法互通，就无法最后构建真正的一个大云。正是基于此，本章就先来讨论在云计算领域的相关开源软件，讨论其历史和优缺点，使得后续大家选择实现云计算产品的时候能做到更好的通用性。

云操作系统开源软件

在云计算领域，开源云计算的软件主要有OpenStack、CloudStack、OpenNebula、Eucalyptus，参与人员的数量和活跃程度、贡献程度又以OpenStack、CloudStack为主。其中OpenStack开源社区由于其架构的开放性和灵活的可扩展性，呈现出后来居上的趋势，参与人员数量和公司都有一骑绝尘的态势（参考蒋清野对开源社区的跟踪研究<http://www.qyjohn.net/?p=3399>）。本书针对开源云计算软件的介绍，重点围绕OpenStack、CloudStack展开，并特别强调作为开源软件，软件架构的开放性、可扩展性对生态系统构建的重要性。

Hypervisor开源软件

Hypervisor领域，既有闭源的ESXi、HyperV，也有开源的Xen、KVM。Xen发展时间长，功能丰富，也得到了广泛的应用，KVM作为Linux内核集成虚拟化技术，则得到了广泛的社区支持，快速发展，也是OpenStack社区最常用的Hypervisor，不少企业和电信运营商更是指定云建设必须基于KVM。

3.2 Cloud OS开源软件：CloudStack

2008年，CloudStack为初创公司VMOps的一个项目，后来公司更名为Cloud.Com，2011年7月，Cloud.com被Citrix收购，其全部源代码都贡献到开源社区。2012年4月，Citrix把CloudStack贡献到Apache作为孵化项目，并于2013年3月正式毕业，成为Apache的一个正式开源项目，软件Licence为Apache 2.0 Licence。

CloudStack面向企业私有云和公有云。作为IaaS的云计算平台，CloudStack兼容多种Hypervisor，如XenServer、VMWare vCenter、Oracle VM、KVM等虚拟机以及裸金属物理机的资源发放；在存储的支持上，基本上覆盖了本地磁盘、iSCSI、FC、NFS等类型的存储设备；网络则支持不同类型的网络隔离，提供防火墙、VPN、负载均衡服务。

3.2.1 CloudStack的总体架构

CloudStack的软件架构如下：系统的主要部分由Management Server和MySQL组成。Management Server通过XenAPI管理XenServer虚拟机集群，通过vCenter API管理vCenter的虚拟机集群，通过CloudStack自己的Agent管理KVM虚拟机集群。通过Management Server对最终用户和管理员提供UI和API服务。

CloudStack提供系统虚拟机用于完成系统管理的作用，比如Secondary Storage VM负责对模板、快照、ISO镜像等进行管理；比如Console Proxy VM用于提供虚拟机的VNC服务。对于租户的网络服务，可以通过网络硬件实现（比如F5、NetScale、Juniper SRX等），也可以通过虚拟机实现（比如Router）。因此Management Server需要对这些系统VM进行管理，同时也要对网络硬件进行管理用于提供虚拟网络服务。

由于CloudStack早期由Cloud.com开发，后又被Citrix收购，系统设计之初对于规模和架构的开放性考虑相对较少在所难免。在成为Apache的正式开源项目之后，则面临着非Citrix和非Cloud.com人员是否能够快速进入开发、降低人员参与社区的门槛等问题，同时随着实际使用和交付的增多，不同厂家的软硬件能力的集成以及系统规模管理能力都时刻考验软件架构是否能够支撑。软件架构成了CloudStack开源社区是否吸引生态系统的参与、是否可持续发展的决定性因素（见图3-1）。

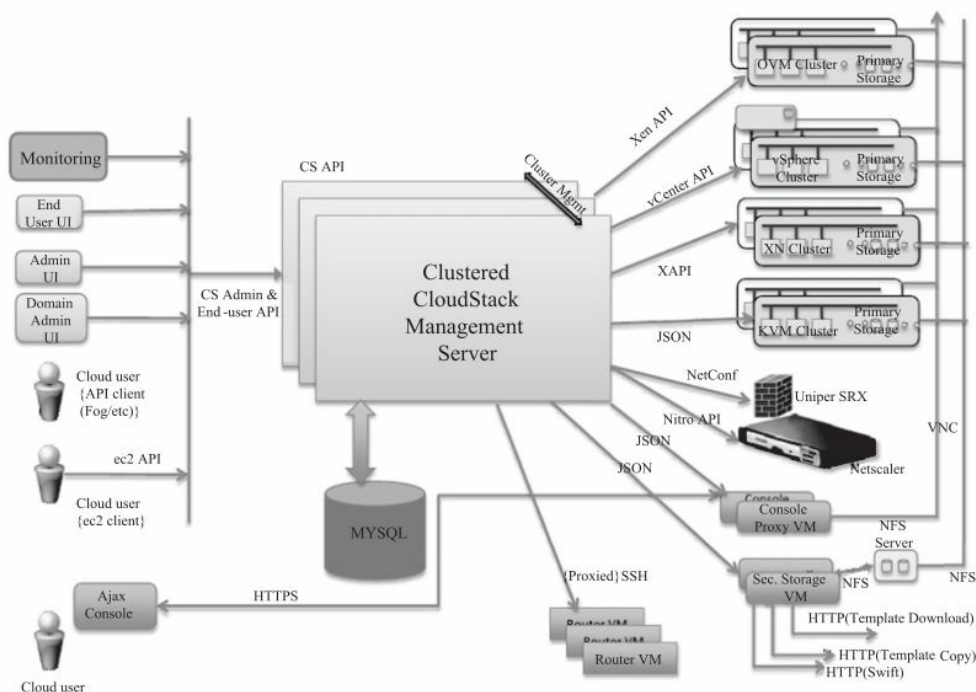


图3-1 CloudStack的软件架构

例如，可能会遇到以下问题。

（1）是否可以增加定制的API？不同集成商针对不同的客户可能有不同的功能要求，是否能够在架构上简单地增加定制化API，而且不需要修改CloudStack源代码，最好也不需要重新编译，使用CloudStack社区原生版本即可做到。

（2）是否可以根据客户的需要，增加调度机制而不需要修改CloudStack的源代码，也不用重新进行CloudStack的编译和发行版本？比如，对于一个由多个虚拟机组成的应用，有的虚拟机需要特殊网卡、特殊的网络虚拟化加速，而另外一些虚拟机和该虚拟机有互斥部署的要求，其中某些虚拟机又必须部署在同一物理主机。

CloudStack贡献到Apache之后，在不断做架构上的重构（见图3-2）。希望从原来紧耦合的Java应用服务器架构修改成基于组件化服务的架构，如把API服务和多资源协同调度分离出来，把Identity管理和权限管理与API分离，业务处理逻辑和数据库分离，对于系统规模增大后资源消耗大的状态管理和计量管理独立为Usage服务，通过这样独立的组件服务，能够为系统带来水平扩展能力。

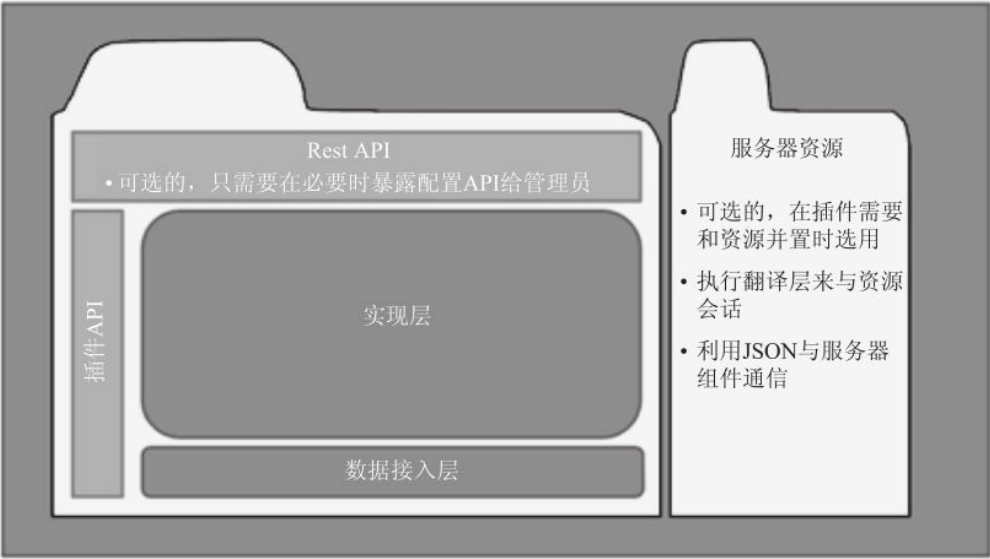


图3-2 基于组件化服务的架构

对于现有系统架构的重构是非常困难的一件事情，不能一蹴而就，比如在4.2版本引入第三方UI的Plugin机制，在最新版本4.3中也开始提供API的Plugin机制，允许增加定制化的API。而且从CloudStack4.3开始，继续对架构进行解耦，Spring框架的使用引入了模块的自动发现及加载。

开发者除了使用系统已经定义好的API，还可以自定义出系统管理用的API。

当前可以开发Plugin的API主要如表3-1所示。

表3-1 开发Plugin的API

API接口名称	说明
NetworkGuru	网络隔离和IP地址管理，主要用于Layer 2的实现
NetworkElement	网络服务（如DHCP、DNS、LB、VPN、Port Forwarding等）
DeploymentPlanner	虚拟机调度器，用来实现不同种类的调度算法
Investigator	主机和虚拟机的状态监控

Fencer	未知状态虚拟机的隔离（比如一个物理主机网络中断，系统判断需要在另外Fencer的物理主机重启原物理主机上的虚拟机，但是后来网络连接又恢复了，就会出现虚拟机状态的冲突。此时就需要Fencer，该功能比较复杂，所以这样的特性不容易稳定）
UserAuthenticator	鉴权认证方法
SecurityCheckerACL	访问控制
HostAllocator	主机分配方法
StoragePoolAllocator	块存储分配方法

图3-3 是展开后Management Server的功能结构图，在独立的服务之间引入Message Bus，同时各种类型的资源通过Event Bus把状态事件信息传送到使用服务器。在API上对外提供兼容EC2的API，分为三种角色授权不同用户以不同API的访问权限：Root Admin、Domain Admin和User。

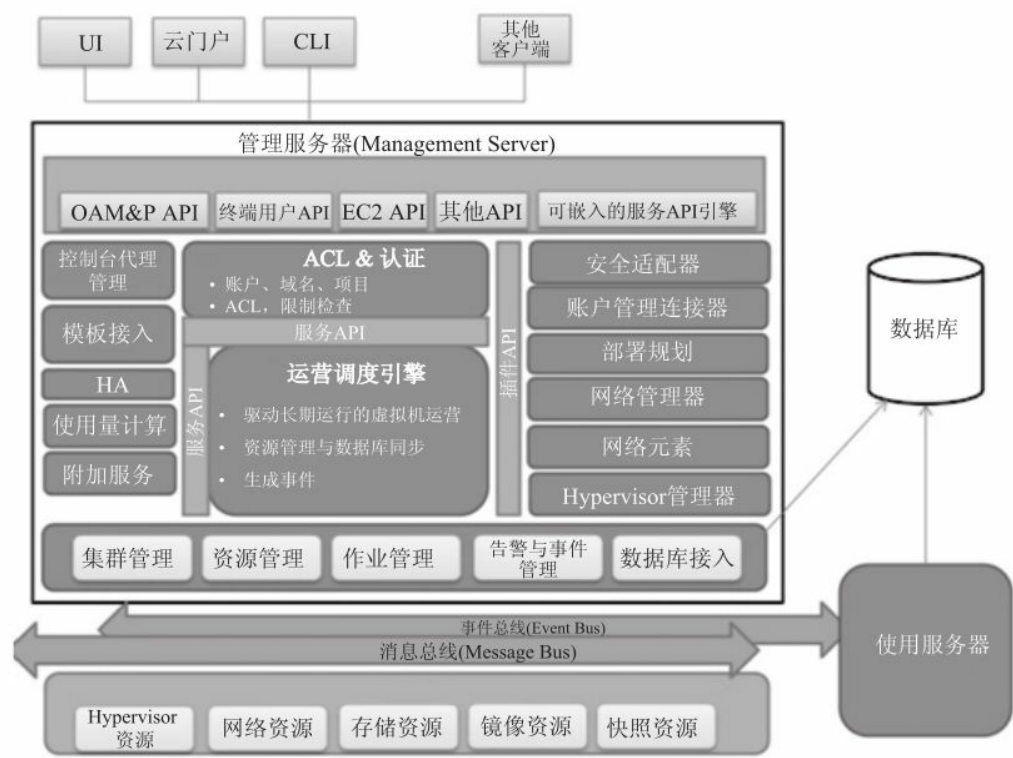


图3-3 Management Server功能结构图

前面谈到CloudStack的系统能力可以通过LB+Management Server集群来水平扩展，这种水平扩展能力是基于Management Server完全无状态，其上层处理逻辑和底层数据库分离之后，通过水平增加Management Server数量来应对系统负荷的增长，水平扩展的极限会出在MySQL数据库上。实际上，在Management Server内还有消息总线（Message Bus和Event Bus），随着主机数量的增加，消息总线的可扩展能力同样可能成为系统的性能瓶颈。处理方式可以通过资源分片，由Management Server只负责相应部分资源的处理，好处是限定了Management Server需要处理的负荷上限，带来的问题则是增加了Management Server之间的交互，在资源扩展的时候以及Management Server的成员管理上都增加了资源重新切分的逻辑复杂性。

3.2.2 CloudStack的资源管理

CloudStack的简化版软件部署架构，如图3-4所示。

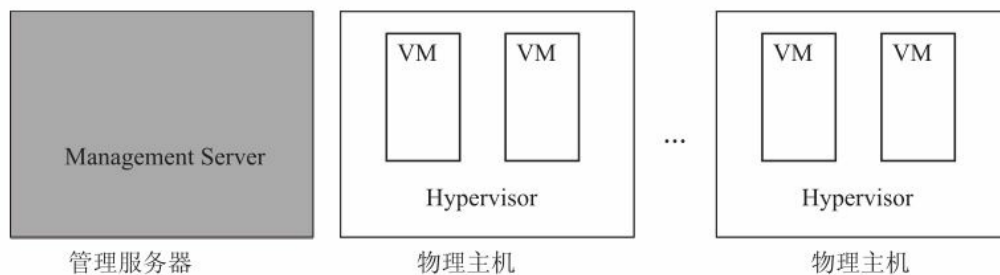


图3-4 CloudStack简化版软件架构

典型地，CloudStack会使用一台服务器作为管理服务器，运行CloudStack的Management Server，用于管理一台或者多台物理主机上的虚拟机。在非生产环境下，也可以在一台服务器上甚至是笔记本的虚拟机上运行整个CloudStack。

Management Server可以运行在独立的物理服务器上，也可以运行在虚拟机上。其主要的功能如下：

- 为CloudStack的管理员提供WebUI，同时也提供API/CLI的访问方式；
- 虚拟机的分配管理；

- 虚拟机存储资源的管理，以及挂载存储资源到虚拟机；
- 提供快照、模板、镜像的管理和复制；
- 网络资源管理，如创建虚拟网络，把虚拟机加入虚拟网络，进行公有或者私有IP地址管理，部署负载均衡，防火墙等。

作为生产环境的云平台，在不同的场合，物理主机的数量极为不同，小到几台物理主机，大到分布在全球各地的多个数据中心。因此云平台必须具备管理不同规模物理主机的能力，但是存储和网络服务的范围不可能无限大，CloudStack对物理主机进行的资源域规模划分，如图3-5所示。

其包括如下级别。

（1） **Host**：物理主机。

（2） **Cluster**：同质物理主机集群，在该集群上，共享存储，虚拟机可以进行热迁移，物理主机同在一个二层网络，使用相同的Hypervisor，一般是一个机架内范围。Cluster范围限制主要取决于共享存储容量大小，因为热迁移要求在共享存储的基础上进行。因为热迁移特性和HA特性（High Availability，一个物理主机坏掉之后，这个物理主机上运行的虚拟机在同一个Cluster中的其他物理服务器上重启运行）有各种诉求，比如必须是相同的Hypervisor、需要虚拟机的系统卷在共享的存储上，这样可以根据需要在不同的物理主机启动这个虚拟机。

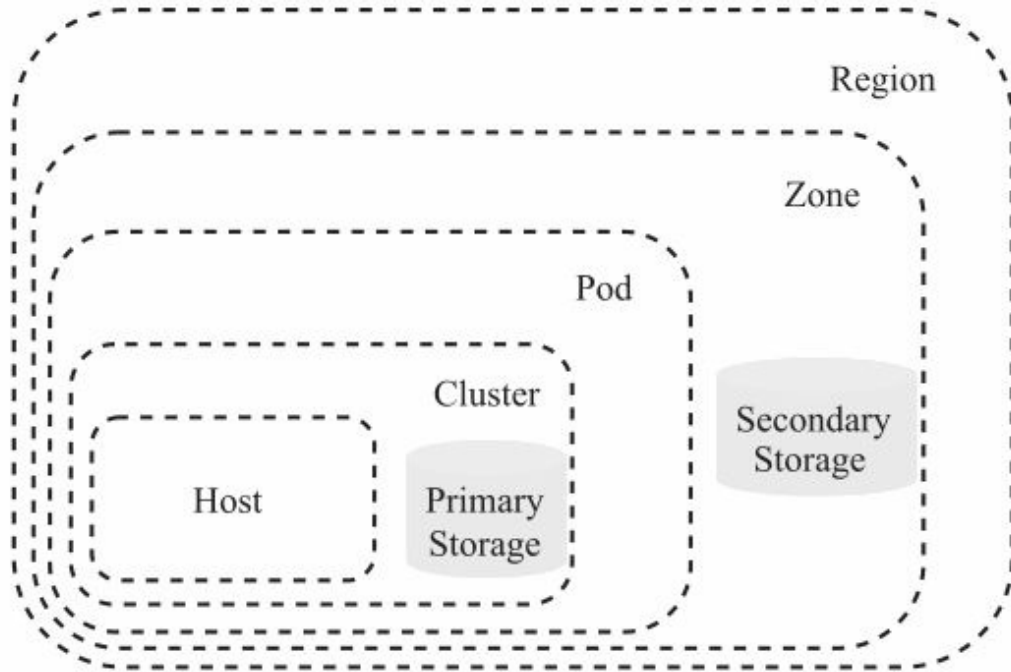


图3-5 CloudStack资源域概念

(3) **Pod**: Pod是比Cluster更大的一个主机集合，Pod范围内的物理主机都同处于一个二层网络内，Pod的范围主要受限于二层网络的范围。Pod内可以是同质的物理主机，也可以不是；Pod内可以划分为几个Cluster，每个Cluster分别有自己的共享存储；当然，在CloudStack里面，并没有限制说，一个Pod内的多个Cluster不能共享同一个存储。如果在Pod内不做虚拟机的热迁移，则Pod内不同Cluster可以是非同质的Hypervisor，也就是每个Cluster可以有自己的Hypervisor。Pod通常是一个机柜，Pod对最终用户并不可见，只对管理员可见。Pod与Pod之间的网络可以是两层互通，也可以是三层互通。

(4) **Zone**: 物理隔离的主机集合，通常指一个数据中心，也常常指一组物理主机的网络边界，Zone之外就是外网了，Zone可以有自己的防火墙、路由器、VPN等服务。Zone的概念等同于Amazon的Availability Zone。Zone之内并不要求是一个二层网络，可以是多个二层网络。对于最终用户来说，Zone是一个可见的资源位置概念，Zone与Zone之间的物理资源和虚拟资源是不共享的，比如虚拟机模板就不共享，也就是说在创建虚拟机的时候，必须显式地指定Zone进行创建。在CloudStack中，一个用户可见的Zone有两种：一种是CloudStack中的Public Zone，对所有租户都是可见的；还有一种Zone，是只对某一个用户集合才可见。

(5) **Region**: 一个Region实际上就是一个CloudStack管辖的范围。多个Region就需要通过其他软件来管理多个CloudStack了。一个Region可以只包含一个Zone，也可以是多个，所以Region范围理论上来说可以是大到多个数据中心，小到只有一个Zone、一个Pod、一个Cluster。

(6) **Primary Storage**: 虚拟机运行需要的系统卷和数据卷都存放在Primary Storage，因为要支持虚拟机热迁移、HA等特性，因此往往在一个Cluster内共享一个Primary Storage，也可以几个Cluster共享同一个Primary Storage，甚至大到一个Zone内的虚拟机共享同一个Primary Storage；但是不会出现一个Cluster内多个Primary Storage的情况，从软件设计来说，要支持跨存储的迁移特性会麻烦一些。CloudStack的Primary Storage支持分级存储，就是当最终用户要创建虚拟机的时候，可以指定在何类存储上创建虚拟机，以满足虚拟机对存储性能的不同要求。不过这意味着选择不同类型的存储，会在不同的Cluster创建虚拟机。当前Primary Storage支持以下类型的存储，但是并不是每一种Hypervisor都支持下面所有类型的存储：

- iSCSI
- NFS
- FC
- Local Storage

(7) **Secondary Storage**: 主要用于存储写入次数少而读取次数多的数据，如虚拟机模板、ISO镜像、磁盘卷快照。Secondary Storage作用范围较大，往往是一个Zone或者一个Region部署一个Secondary Storage。CloudStack主要支持NFS存储作为Secondary Storage，当前也在提供Plugin用于接入OpenStack的Swift或者AWS的S3。

针对上述资源划分的方法，CloudStack的物理资源一般如下设计，在一个Pod内包含几个Cluster，每个Cluster有数个物理服务器，Pod通过二层交换机进行汇集，接入到3层核心交换机。一个Zone内共享一个Secondary Storage，一个Cluster共享一个Primary Storage（见图3-6）。

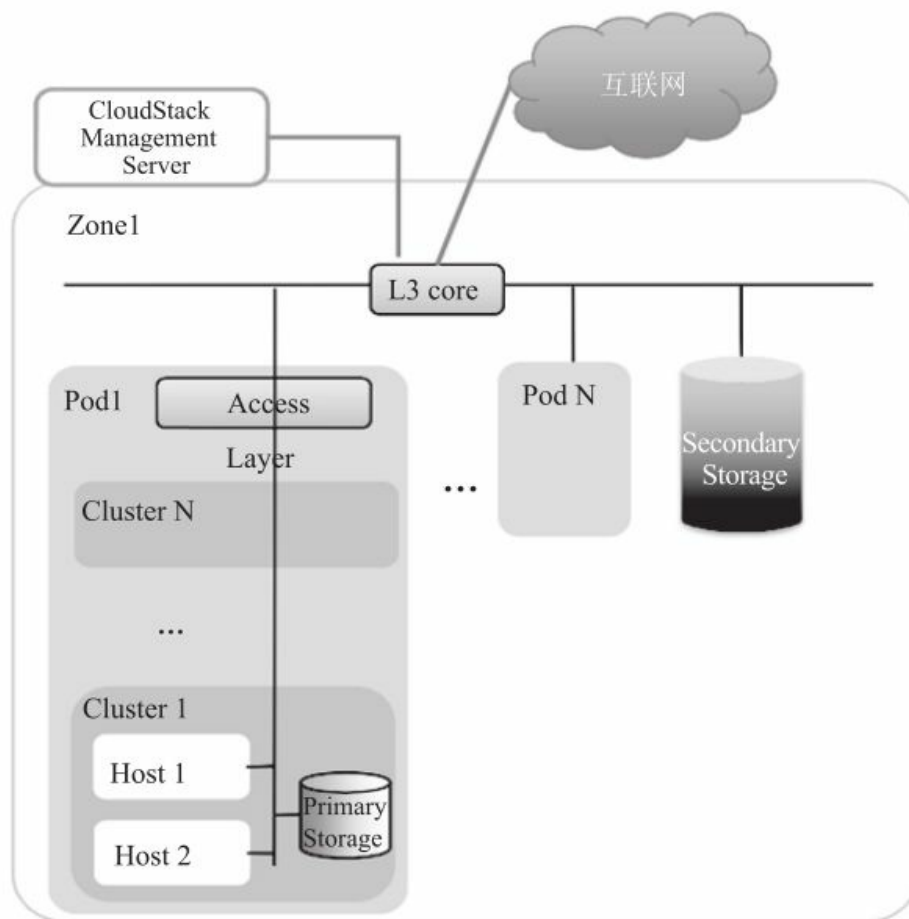


图3-6 CloudStack概念示意图

作为一个面向商用交付的云管理软件，需要兼容多种不同的Hypervisor，这些Hypervisor可能同时部署，也可能一种生产环境只部署一种Hypervisor。因此，Management Server需要能够支持异构多Hypervisor。对于VMware的接入，CloudStack主要是通过vCenter的API来进行管理，其高级特性，如HA、LiveMigration等实际由vCenter来实现，因此较为成熟；同样地，对于整个CloudStack可以管理的系统规模来说，因为多了一层vCenter管理层，其规模可以较容易达到更大的规模，比如1万物理主机。而对于KVM，由于没有vCenter这样的虚拟化管理层，是通过在KVM主机部署一个Agent，Agent再去调用Libvirt的接口来进行的，因此一些vCenter有的高级特性（DRS、FT、HA、Live Migration等）实现就相对晚一些，也没有这么成熟。CloudStack要进行KVM的管理，由于KVM本身没有Cluster的概念，也缺少vCenter的这样虚拟化管理层，因此其管理功能都集中到了CloudStack的Management Server，因此针对KVM的整体系统规模很难达到接入vCenter同等的系统

规模（见图3-7）。

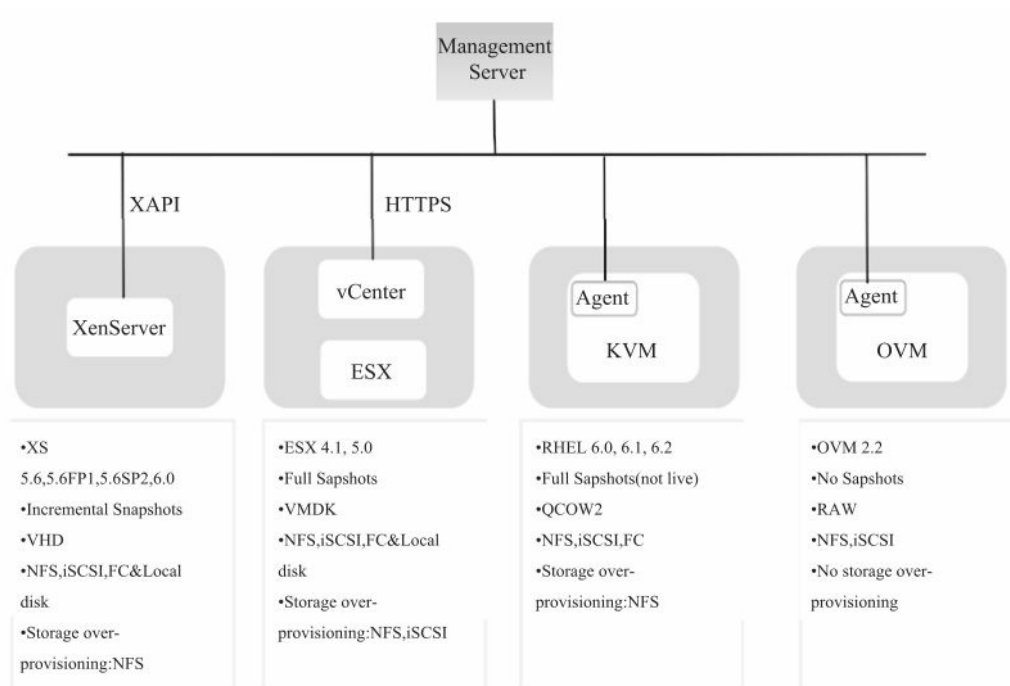


图3-7 CloudStack兼容多种不同的Hypervisor

在大规模的生产环境中，一般至少是在一个Pod或者Zone级别采用同一种Hypervisor，在一个Pod内的不同Cluster采用不同的Hypervisor可能性非常小，很多场景是在整个CloudStack的管理域只采用一种Hypervisor，无论对于管理还是维护工作都会相对简单。

通过Management Server的Cluster部署可以管理更大的规模（见图3-8）。由于当前Management Server主要采用MySQL作为数据存储的后端，而MySQL的Cluster并不是非常成熟和好用，因此Management Server当前的规模主要是通过扩展多个Management Server来进行的，在多个Management Server上提供负载均衡，数据库则采用主备方式提供可靠性，系统的瓶颈最终存在于数据库的性能。当然CloudStack可以把后端数据存储改为其他提供商用集群能力的数据库，但是成本就上去了。

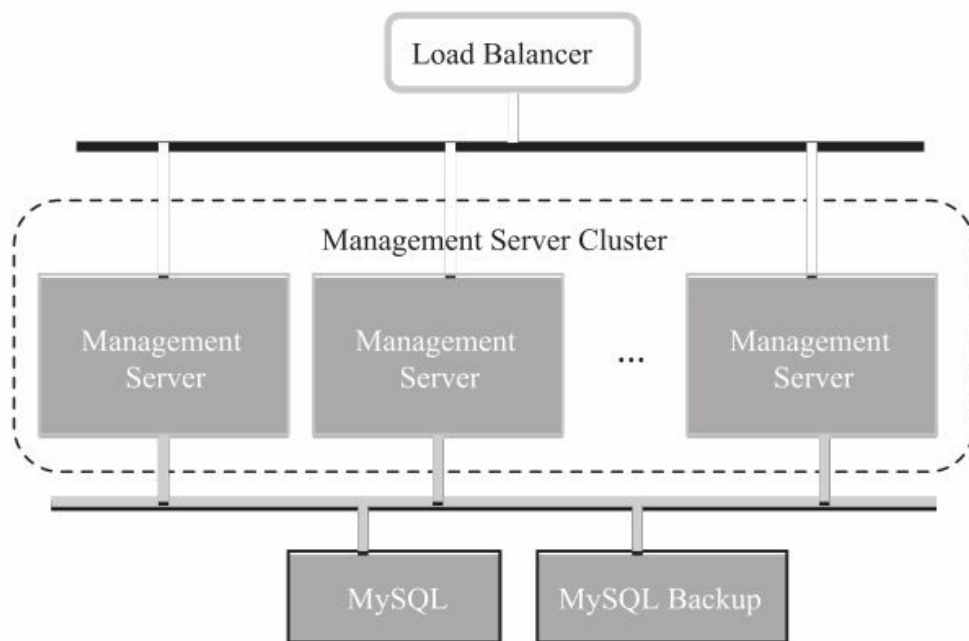


图3-8 Management Server的Cluster部署

CloudStack可以在一个Region内进行多数据中心的管理，一个数据中心可以部署一个到多个Zone，多数据中心的管理主要是管理这些数据中心的Zone，Zone之间要求用低时延网络互联（见图3-9）。CloudStack Management Server需要和部署在物理主机的Agent或者vCenter/XenServer虚拟化管理软件通信，因此跨数据中心管理网络必须联通且管理网络为内网，也可以把vCenter直接对应到CloudStack的Zone概念上。

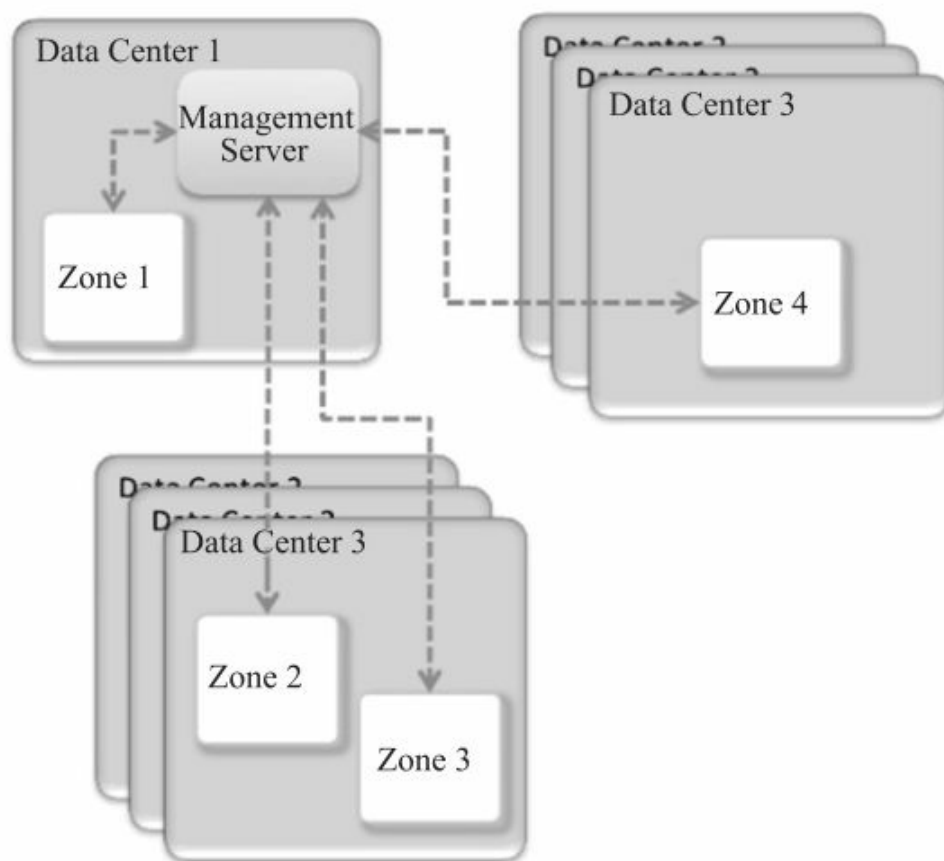


图3-9 CloudStack多数据中心管理

从理论上来说，要求Management Server可以跨数据中心管理Zone是可以的，但是实际部署的时候需要考虑系统的健壮性，比如：如果部署Management Server的数据中心不可访问，则所有数据中心的资源都处于不可管理状态，因此一般会要求把Management Server部署在两个数据中心，数据中心间数据库MySQL做实时备份。

[3.2.3 CloudStack的虚拟机管理](#)

CloudStack提供基本的虚拟机管理能力，包括虚拟机的生命周期管理（创建、启动、停止、重启、删除），以及通过VNC远程访问虚拟机，查看虚拟机状态及虚拟机弹性伸缩（见图3-10）。



图3-10 CloudStack虚拟机管理能力

3.2.4 CloudStack的块存储管理

CloudStack提供虚拟机块存储管理，可以为虚拟机做卷的CRUD，以及从卷创建模板和快照进行卷快照和管理（见图3-11）。



图3-11 CloudStack块存储管理

3.2.5 CloudStack的虚拟网络

一个典型的Zone内的物理组网如下，Management Server和数据库MySQL部署在一个管理网络，所有Pod内的物理主机和存储也都需要连接到这个管理网络，这样Management Server、vCenter/XenServer/Agentc、存储管理软件之间才能够通过消息总线进行通信。事件和计量数据也可以通过管理网络汇总到UsageServer。在管理网络上部署Secondary Storage，用于存储虚拟机模板、快照、ISO镜像等（见图3-12）。

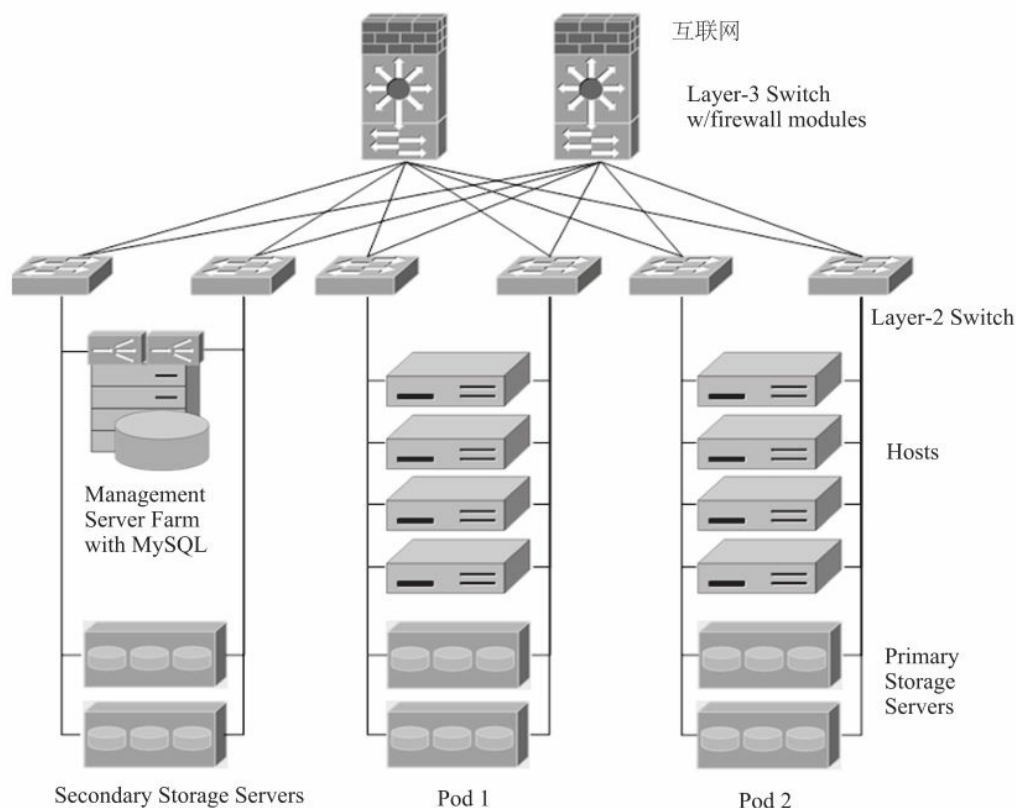


图3-12 Zone内物理组网

如下为一个Pod内的典型物理网络连接。Pod交换机一般会是一对，所有Pod内的物理主机都由上行口连接到两个Pod交换机，交换机的上行口推荐10Gb带宽。物理主机和存储都连接到管理网络，同时物理主机还要配置租户用的一个或者多个网络。如果租户网络需要连接外网，则物理主机也需要连接到一个外网网络（见图3-13）。

对于CloudStack的租户来说，其租户网络只能在一个Zone内，Zone之间的同一个租户的网络只能通过路由的方式访问。

CloudStack给最终用户提供三种类型的虚拟网络（见图3-14）。

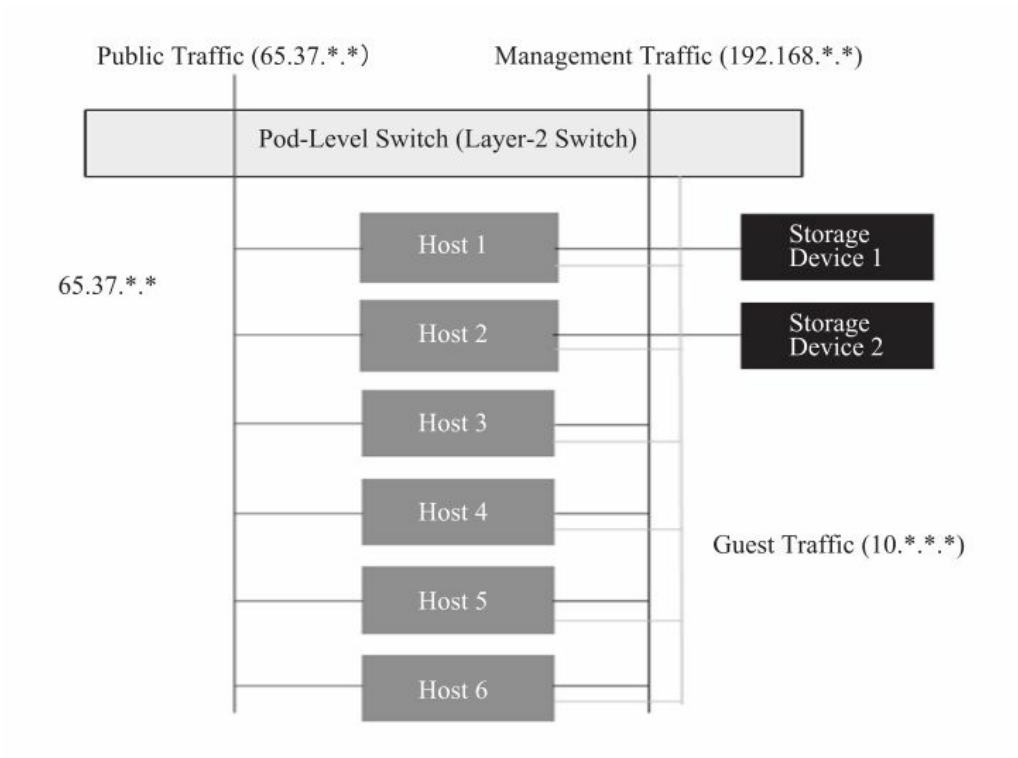


图3-13 Pod内典型物理网络连接

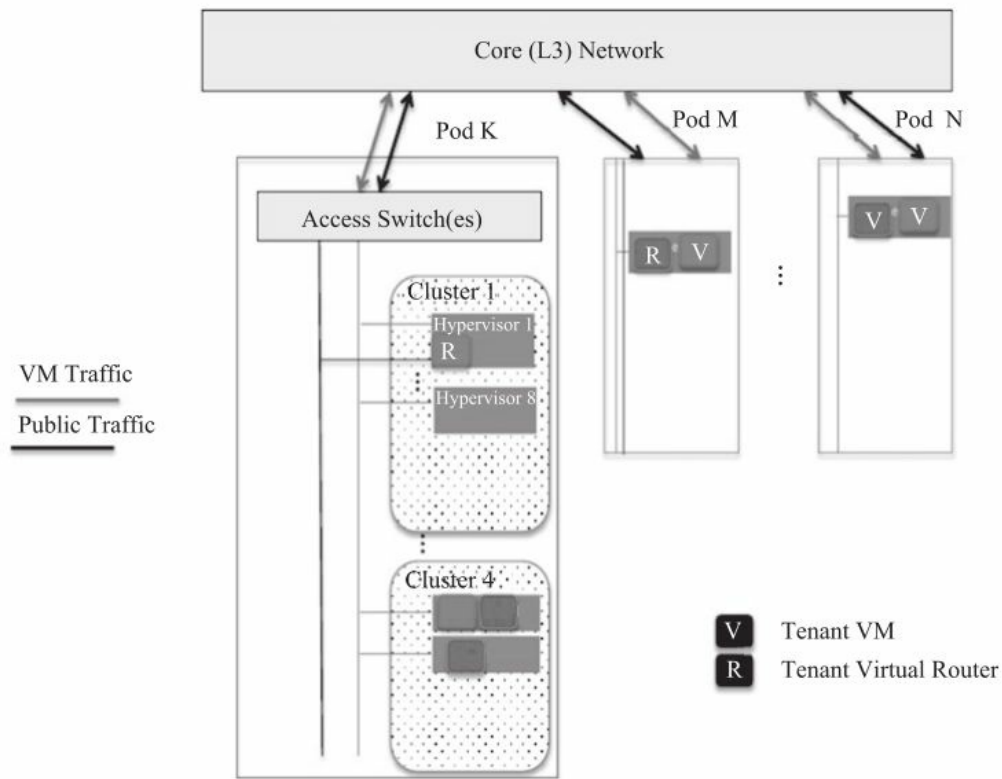


图3-14 CloudStack的多租户虚拟网络

(1) 共享网络：所有不同账户的虚拟机都在一个共享网络内，虚拟机之间通过安全组进行隔离。

(2) 隔离网络：为租户提供二层隔离的虚拟网络。租户内的虚拟机可以分布在不同的Pod内。

图3-15示例了Guest1和Guest2两个隔离网络，IP地址可以重叠。

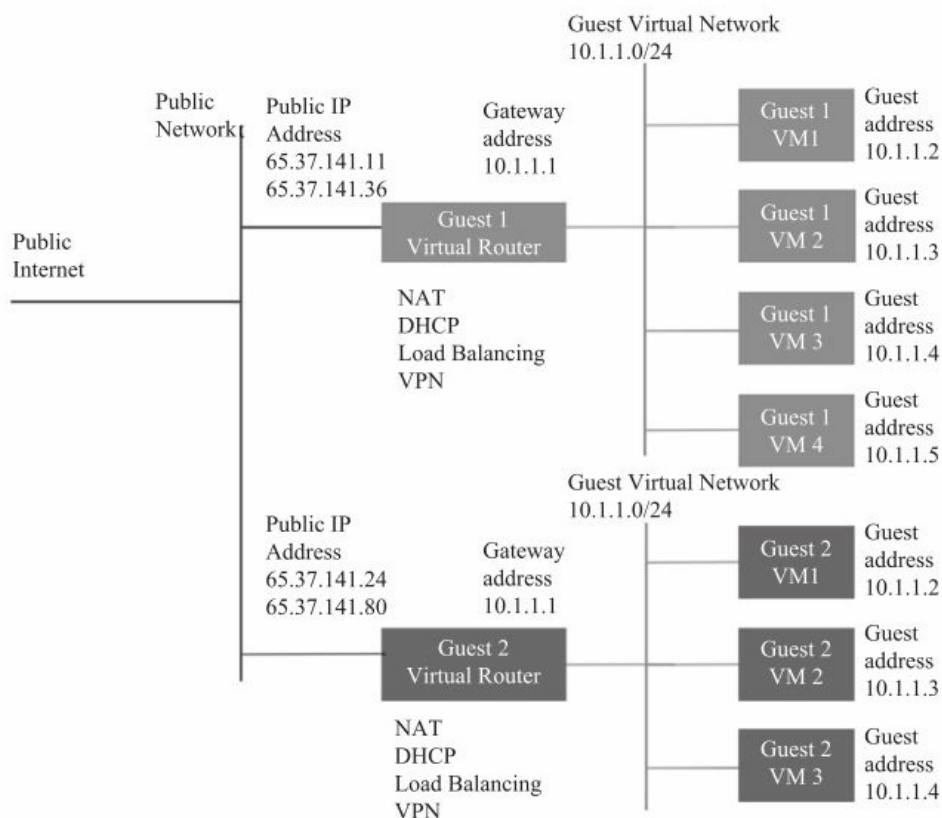


图3-15 IP地址可以重叠的虚拟网络

(3) 运行时网络：所有定义的虚拟网络资源，只有当第一个虚拟机加入这个网络的时候才进行创建，当最后一个虚拟机删除的时候，虚拟网络资源也被回收。

CloudStack可以为虚拟网络配置网络服务，包括：DHCP、DNS、源地址NAT、静态NAT、端口转发、负载均衡、防火墙、VPN、为指定使用的某种网络设备提供网络服务、指定使用某个物理网络。CloudStack默认认为虚拟网络创建一个系统VM，包含DHCP、SNAT、Router三个功能，确保租户网络内的虚拟机自动分配IP，能够通过SNAT和Router访问外网。

CloudStack也支持复杂多层应用的虚拟网络，图3-16为一个典型的多层Web应用，应用分为数据库层、App层和Web层，每一层都有多个虚拟机，每一层通过CloudStack的Virtual Router虚拟机提供网关功能，Web层服务和外网之间部署防火墙和Load Balance网络服务。

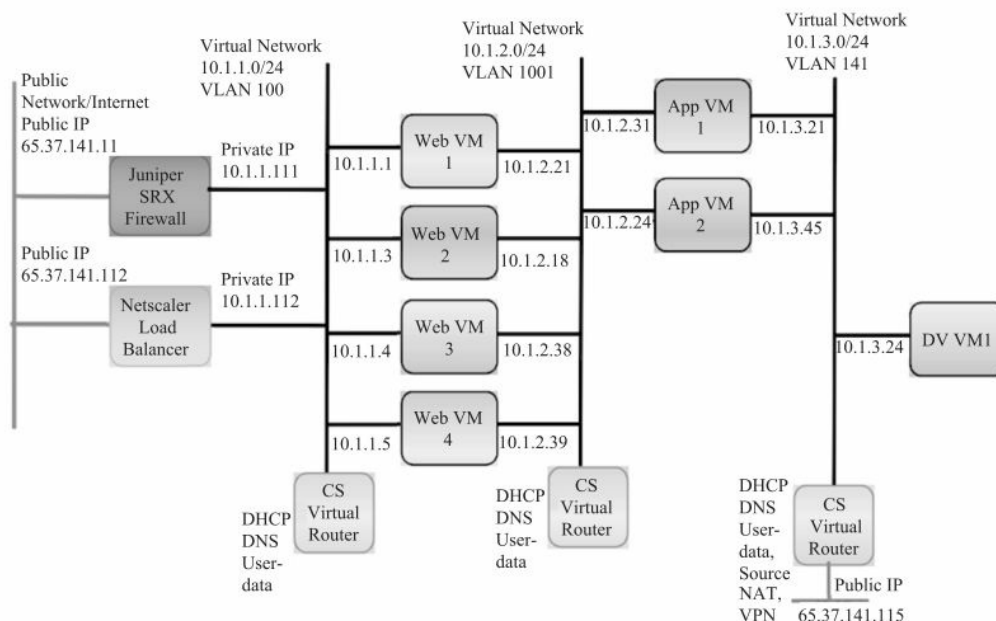


图3-16 CloudStack支持复杂多层应用虚拟网络

3.3 Cloud OS开源软件：OpenStack

2010年7月，NASA与RackSpace一拍即合，NASA贡献Nova，RackSpace拿出Swift，创立OpenStack。经过三年的发展，OpenStack已经成为当前最炙手可热的云计算开源社区。

OpenStack是一个旨在为公有云及私有云的建设与管理提供软件的开源项目。从2012年推出到2014年初，OpenStack已经吸引了超过190家公司和超过15 000名开发者。让它在短时间内声名远播的是其拥有着IBM、HP、AT&T、Red Hat、SUSE、Canonical、Cisco、Dell、VMware、华为这样的强力支持者。

越来越多的企业乐于使用那些不是以单一厂商主导的云计算基础设施，开源软件项目可以帮助他们摆脱对某类产品的强大依赖，无论从开放性、灵活性还是成本上来说，开源软件都是一个非常好的选择。

OpenStack秉承开放的理念，获得了用户和开发者的广泛认同，已成为业界最有影响力和发展前景的云计算开源项目。

3.3.1 OpenStack的总体架构

OpenStack是一个松耦合的架构，由一组离散的服务组成（见图3-17、表3-2）。

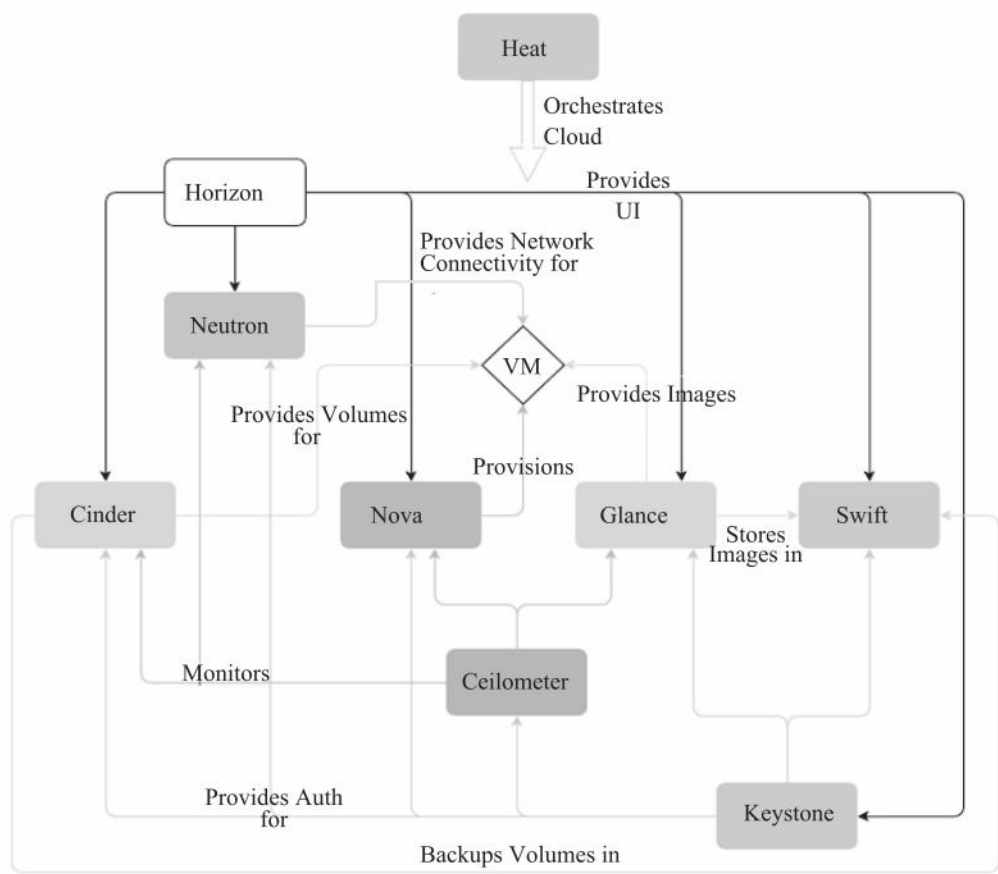


图3-17 OpenStack的总体架构

表3-2 OpenStack由一组离散的服务组成

服务名称	服务名	描述
GUI	Horizon	提供Web GUI与各个OpenStack服务进行交互，使得用户能够进行虚拟机管理、卷管理、虚拟网络管理、分配IP地址、挂载卷、虚拟机加入虚拟网络、设置访问控制策略等
身份服务	Keystone	向所有其他的OpenStack服务提供鉴权和认证服务，提供基于RBAC的用户管理、服务目录管理

计算服务	Nova	虚拟机发放和管理服务，通过Driver机制，支持绝大多数当前已知的Hypervisor
对象存储服务	Swift	类似Amazon S3的文件存储和访问服务
块存储服务	Cinder	向虚拟机提供持久化块存储服务
镜像服务	Glance	提供虚拟机的镜像管理，Nova必须依赖虚拟机镜像来启动虚拟机
网络服务	Neutron	用于创建和管理虚拟网络，向OpenStack其他服务提供网络连接即服务的能力；可以在虚拟网络创建虚拟端口，然后把虚拟机插入到这个虚拟的网络端口上；Neutron具备插件化的架构以支持现实世界林林总总的网络设备和技术
计量监控服务	Ceilometer	提供OpenStack的监控和计量服务，用于计费、统计、标杆比对、弹性伸缩等
编排服务	Heat	提供HOT（Heat Orchestration Template）模板和兼容AWS CloudFormation模板的应用编排引擎；既提供OpenStack的API，也提供AWS兼容的API

实际上，OpenStack一开始的架构并不是如图3-17所显示这样的，也是一步一步快速演进过来的。OpenStack基本上保持每半年发布一个版本的节奏，在快速进化中不断优化架构和增加服务能力，表3-3展示了OpenStack的进化能力。

表3-3 OpenStack的发展

版本	发布时间	包含的服务
Austin	2010年10月	Nova, Swift
Bexar	2011年2月	Nova, Glance, Swift

	月		
Cactus	2011年4月	Nova, Glance, Swift	
Diablo	2011年9月	Nova, Glance, Swift	
Essex	2012年4月	Nova, Glance, Swift, Horizon, Keystone	
Folsom	2012年9月	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder	
Grizzly	2013年4月	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder	
Havana	2013年10月	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer	

OpenStack社区把项目分成三类，能够在一个版本中发布的项目称为Core Project（核心项目）。要进入核心项目，首先进入到孵化项目列表中，经过孵化、评估后，达到核心项目的要求才会批准进入到下一个版本进行发布（见图3-18）。任何人都可以递交想法和项目到OpenStack社区，由社区评估认为将来OpenStack发展必须要有的能力，会批准进入孵化项目阶段。

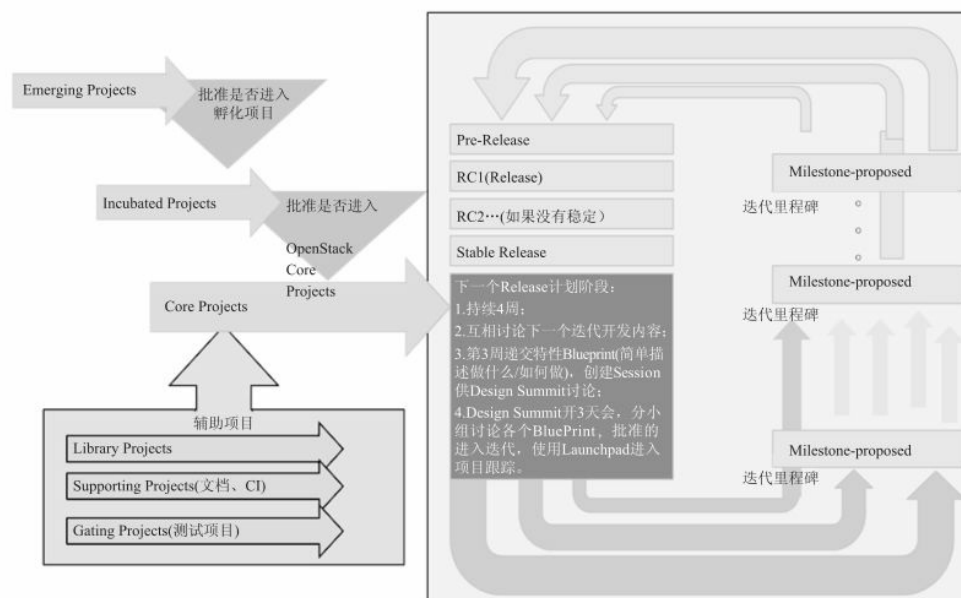


图3-18 OpenStack的运作过程

每个正式发布的核心项目，在上一个版本即将结束、下一个版本开始前的4周内收集各种需求、创建设计和讨论，然后进入下一个版本的开发，一般一个版本分成3个里程碑，以6个月一个版本的节奏不断滚动向前。

OpenStack能够快速进化，除了社区参与人员和厂家众多，与架构也有着莫大的关系。OpenStack的架构设计遵循这些原则：

- 伸缩性和弹性是我们的主要目标；
- 任何会约束我们目标的特性都是可选的；
- 一切都应该是异步的，如果不能异步，那么参考第2条进行设计；
- 所有必选服务部件必须是可水平扩展的；
- 总是使用Shared Nothing的架构或者Sharding技术（切分技术），如果做不到Shared Nothing或者Sharding，那么参考第2条进行设计；
- 一切都应是分布式的，特别是处理逻辑；
- 接受最终的一致性，尽可能地贯彻这一原则；
- 测试一切，所有递交的代码都应进行测试。

下面我们通过典型服务部件来看OpenStack为什么能够吸引如此多的人参与，并快速进化。

3.3.2 OpenStack的计算服务：Nova

Nova的架构如图3-19所示。

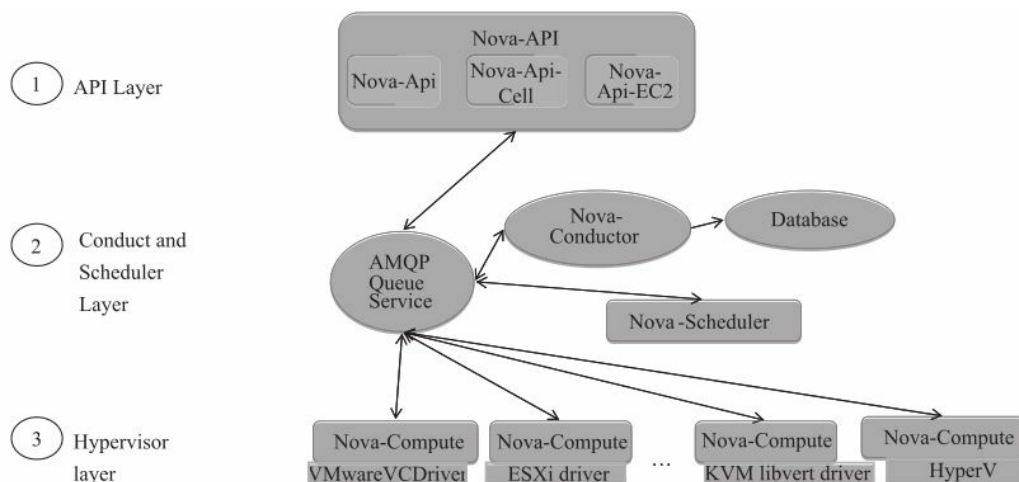


图3-19 Nova的架构

各模块功能如表3-4所示。

表3-4 各模块功能

模块	功能
Nova-API	提供NOVA服务的OpenStack API接口
Nova-API-Cell	以Cell方式大规模部署Nova服务时，Nova提供的API接口
Nova-Cells	提供Cells级联功能，主要作用是以Cells为单元提供超大规模的能力，当部署了Cells服务的时候，Cells服务完成Cell这一层的调度
Nova-Scheduler	是一个或多个AZ（Availability Zone）的调度器，用于在一个AZ或多个AZ中寻找符合创建虚拟机条件的物理主机。一个Cells中可以包含多个AZ
Nova-Compute	在每一个物理主机中都部署Nova-Compute服务，此服务用于该物理主机节点的虚拟机操作
Nova-Conductor	Nova-Conductor主要是屏蔽掉数据库的操作，使得每一个部件不需要直接和数据库打交道，数据库可以水平扩展而不影响其他服务访问数据库的能力
AMQP	模块之间的消息总线，社区用RabbitMQ的比较多

下面以Grizzly版本的源代码分析，从创建虚拟机的过程分析其源代码就可以看出Nova在架构和实现上的开放扩展能力。Grizzly版本创建虚拟机流程和Havana及之后版本有所不同，Grizzly版本是从Nova-API直接到Nova-Scheduler服务，而Havana版本则先到Nova-Conductor，Nova-Conductor再调用Nova-Scheduler进行调度，但是并不妨碍我们对Nova开放性及扩展能力的理解。

在源代码中，首先请求被传递到Servers.py这个脚本文件的Controller类的create函数，这个函数把消息参数全部解析出来，然后传送给被调用的函数。创建虚拟机的请求携带了几个参数，在所有的函数调用和处理中都会一路传递下去，可以用于自定义扩展能力，如user_data、metadata、injected_files、scheduler_hints（见图3-20、图3-21）。

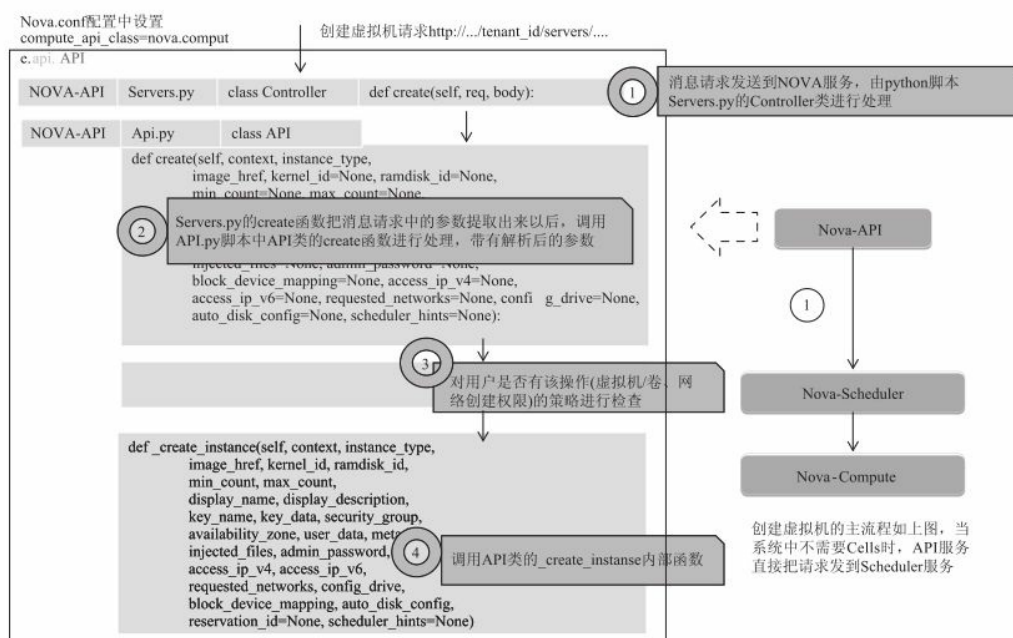


图3-20 Nova API创建虚拟机的过程1

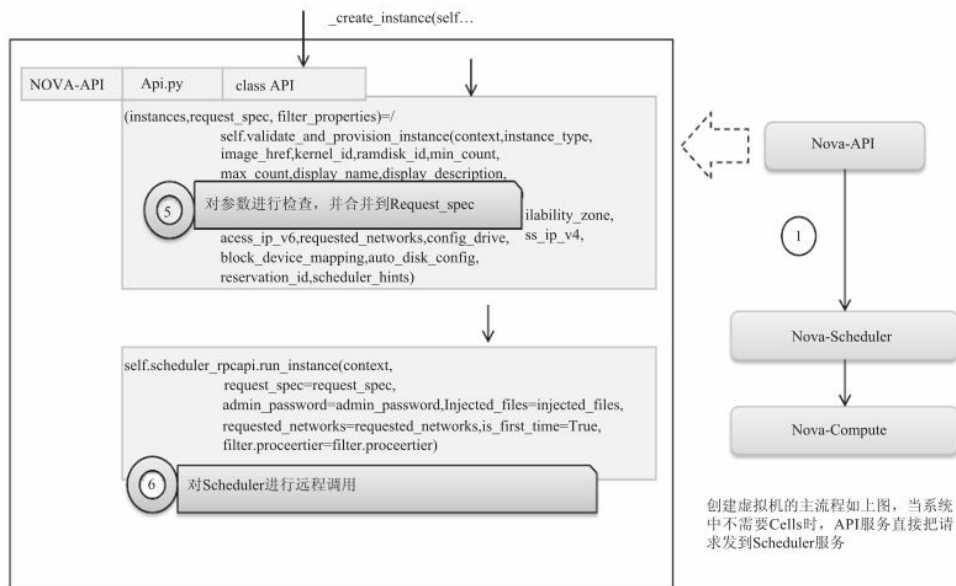


图3-21 NovaAPI创建虚拟机的过程2

Nova-API对参数进行检查合并等处理后，调用Nova-Scheduler去选择可以分配虚拟机的主机。如果有Nova-Cells服务，创建虚拟机的请求会先到Nova-API Cell，然后通过Nova-Cells服务选择Cell之后，再传递到Nova-Scheduler（见图3-22）。

Nova-Scheduler接受到请求之后，会通过scheduler_driver_opt选项获得配置的具体调度器Driver是哪一个，默认的调度器Driver是nova.scheduler.filter_scheduler.FilterScheduler，这个默认的调度器是可以被替换的。

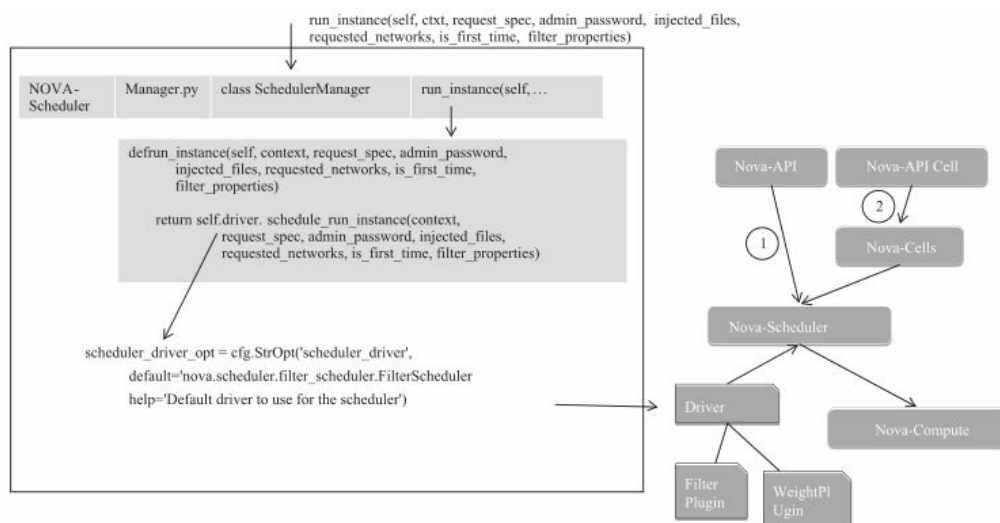


图3-22 可配置的Nova Scheduler

通过分析源代码，我们可以看到调度器有以下定制方法（见图3-23）。

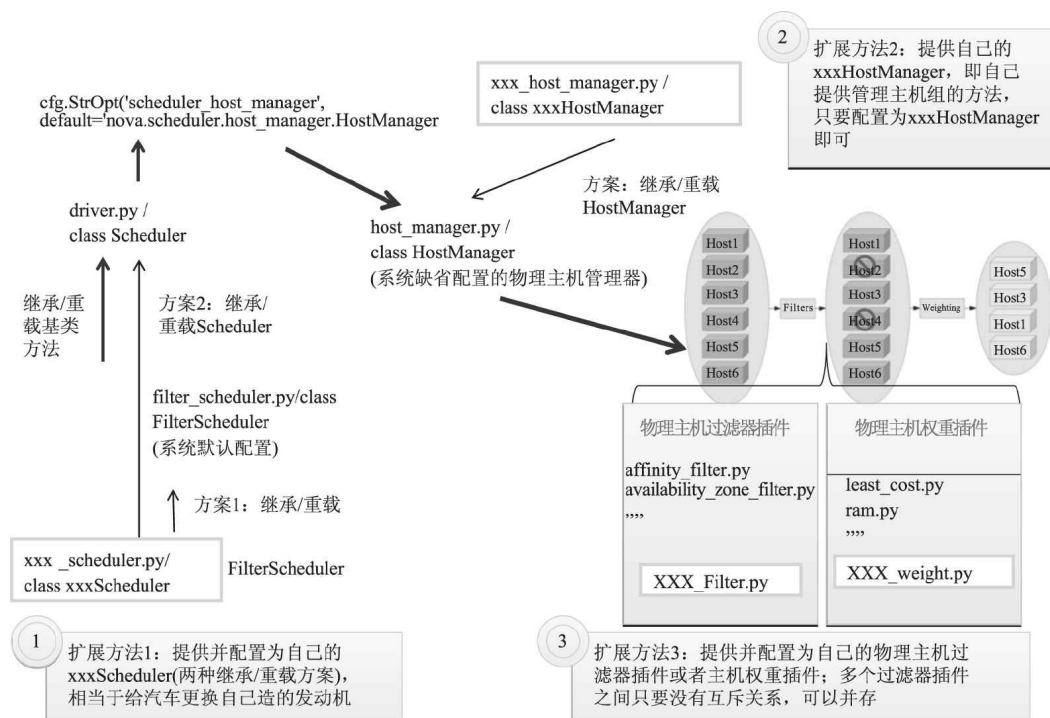


图3-23 Nova Scheduler的可扩展机制1

第一种扩展方法是不用默认的`filter_scheduler.py`的`class FilterScheduler`，自己完全可以从`driver.py/class Scheduler`继承并实现自己的类。

第二种扩展方法是不使用默认的物理主机状态管理类 `host_manager.py/class HostManager`，自己从`HostManager`继承或者重载实现`xxxHostManager`类进行物理主机状态的管理。

第三种方法是提供自己的调度过滤器或者权重过滤器插件，参考 `affinity_filter.py`或`ram.py`实现自己的插件，在部署的时候配置为自己的插件即可。

除了上面的三种方法，调度器还在各种地方提供了扩展性。

例如第四种方法，其在物理主机中定义一个主机组Group，然后在 `Scheduler Hint`中带上这个Group，也可以实现可定制的调度策略。

第五种方法是在Scheduler Hint中增加自定义的调度条件，在调度的时候通过自己的Filter插件增加调度能力。

第六种方法是在创建虚拟机请求的AZ参数中通过“AZ: host”的方式指定主机进行虚拟机的创建。

第七种方法是调度器始终监控一个调度配置json文件，如果这个文件发生变化，则动态加载这个调度配置文件，改变和调整调度策略，因此可以动态修改这个json文件修改调度策略（见图3-24）。

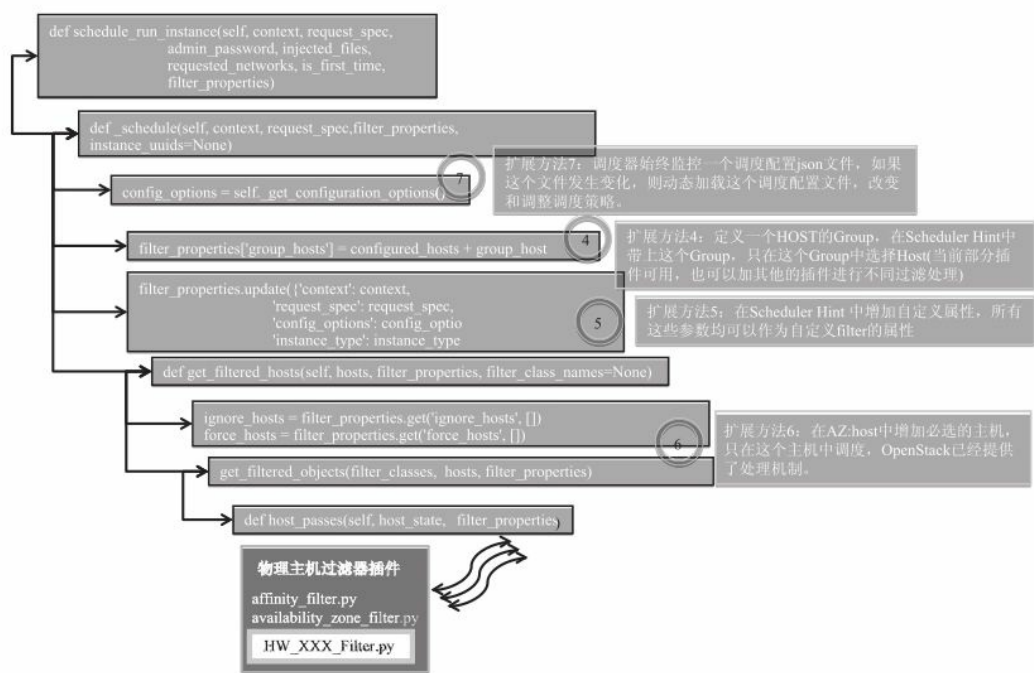


图3-24 Nova Scheduler的可扩展机制2

从NOVA Scheduler上述机制我们可以看出，OpenStack为系统的开放性和扩展型引入各种可能的扩展方案，如driver/plugin/静态配置/动态配置/参数自定义。OpenStack本身就已提供了多种调度过滤器，并在不断丰富中，这些过滤器只要不是互斥的，均可以组合使用（见表3-5）。

表3-5 各插件的功能和使用方法

插件名称	功能	使用方法
------	----	------

CoreFilter	设置主机能够分配虚拟机的vCPU/pCPU的比例，超过比例就不能分配虚拟机了；允许设置为超分配	全局配置数据
AggregateCoreFilter	针对某个Host Aggregate单独设置主机能够分配虚拟机的vCPU/pCPU的比例，超过比例就不能分配虚拟机了；没有设置，默认就用全局的CoreFilter的配置数据	针对HostAggregate设置metadata（Key,Value），置cpu_allocation_ratio=10
AggregateInstanceExtraSpecsFilter	在指定的HostAggregate中选定一个主机分配虚拟机	需要在HostAggregate打上的对应标签，就是HostAggregate和flavor要相同的metadata（key,value）如两个主机分配虚拟机个storage=DSWare的key-value metadata，则表明只在存Dsware的主机分配虚拟机flavor的虚拟机
AggregateMultiTenancyIsolation	在指定的HostAggregate中只允许给某个租户分配虚拟机	HostAggregate增加metadata（key,value）filter_tenant_id=xxx
RamFilter	和CoreFilter类似，只不过针对的是内存	参考CoreFilter
AggregateRamFilter	和AggregateCoreFilter类似，只不过针对的是内存	参考AggregateCoreFilter
AllHostsFilter	允许所有的HOST参加分配	

AvailabilityZoneFilter	只允许在创建虚拟机参数中带的Availability Zone内的那些Host上分配虚拟机
ComputeCapabilitiesFilter	用于过滤满足Flavor的extra_specs条件的AggregateInstanceExtraSpecs主机就要用到extra_specs
ComputeFilter	所有能用能操作的主机都可以分配虚拟机 默认需要配置该过滤器
DiskFilter	根据磁盘分配比例过滤能够分配的虚拟机，超过比例就不能分配了（允许配置为超分配） 在nova.conf disk_allocation_ratio=1.0
DifferentHostFilter	指定不要和某些虚拟机实例共享主机，可以用于互斥的分配场景下 在Scheduler hint中增加hint‘os:scheduler_hints’: {‘different_host’: [‘vm1’, ‘vm2’]}
GroupAntiAffinityFilter	指定不要在某个组内的所有HOST上分配虚拟机，可以用于互斥的分配场景下 在Scheduler hint中增加hint‘os:scheduler_hints’: {‘host1’, ‘host2’}]
ImagePropertiesFilter	根据镜像属性过滤主机，主要的属性包括CPU架构，Hypervisor的类型，VM_Mode[Hypervisor application binary interface（ABI）]等 glance image-update image property architecture=arm64 property Hypervisor_type=qemu
IsolatedHostsFilter	在管理员指定的一些孤立的主机中创建特定镜像的虚拟机 需要在nova.conf配置at_e_d _hosts=server1 isolated_images=342b492c4a42-8d3a-c5088cf27d13,ebd267a6-ca

		4d6c-9a0e-bd132d6b7d09
JsonFilter	在Schedluer hint中带上json脚本，选出满足脚本运算规则的主机分配虚拟机	如设置这样的运行 os:scheduler_hints': { ['>=",\$free_ram_mb",10
RetryFilter	不再对已经尝试分配虚拟机但是失败的主机再次进行选择，只有当nova.conf中允许调度失败时重新调度的配置有效时，这个指标才起作用	scheduler_max_attempts>
SameHostFilter	在指定虚拟机所在主机上分配虚拟机	scheduler hint中加上sar 的设置'os:scheduler_hints' 'same_ host': ['a0cf03a 4877-bb5c- 86d26cf818e1','8c19174f-4 44f0-824a-cd1eeef10287'],
SimpleCIDRAffinity Filter	根据某个子网IP地址范围内的主机进行虚拟机分配	如下为在192.18.1.1的子 虚拟机'os:scheduler_hints' {'build_near_host_ip': '192.168.1.1','cidr': '24'}

当调度器选择合适的主机进行虚拟机创建的时候，命令会来到Nova Compute（见图3-25）。

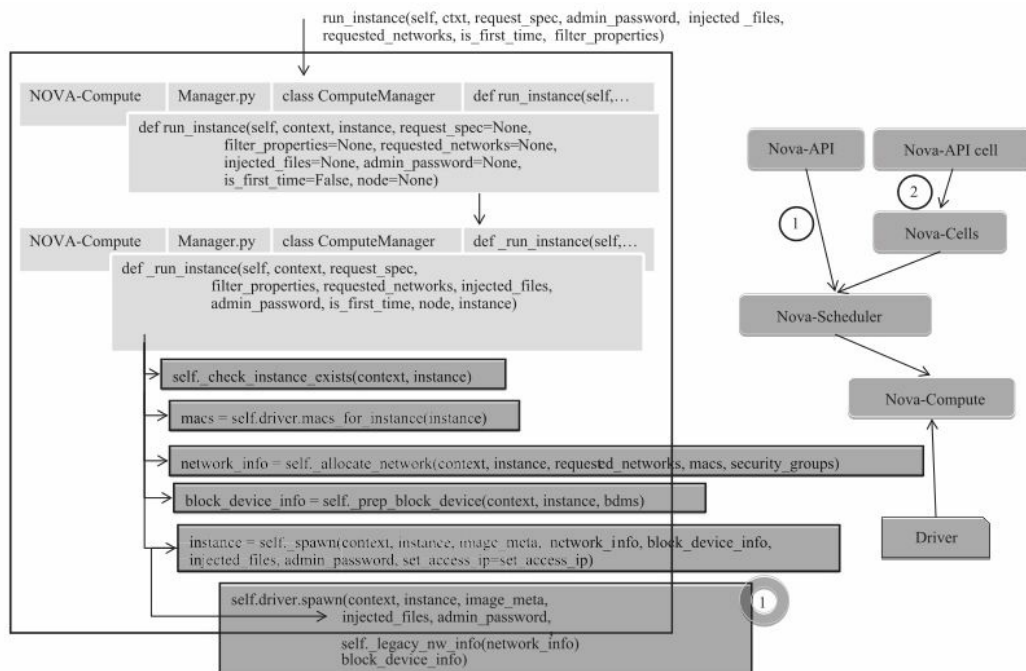


图3-25 Nova Compute创建虚拟机的过程

Nova-Compute采用了Driver机制，使得不同类型的Hypervisor在OpenStack下实现同样的功能。每个厂家只需要根据Driver机制，对类进行继承和重载，针对相应Hypervisor的实现类的方法即可。图3-26对Driver机制已经解释得比较清楚了。

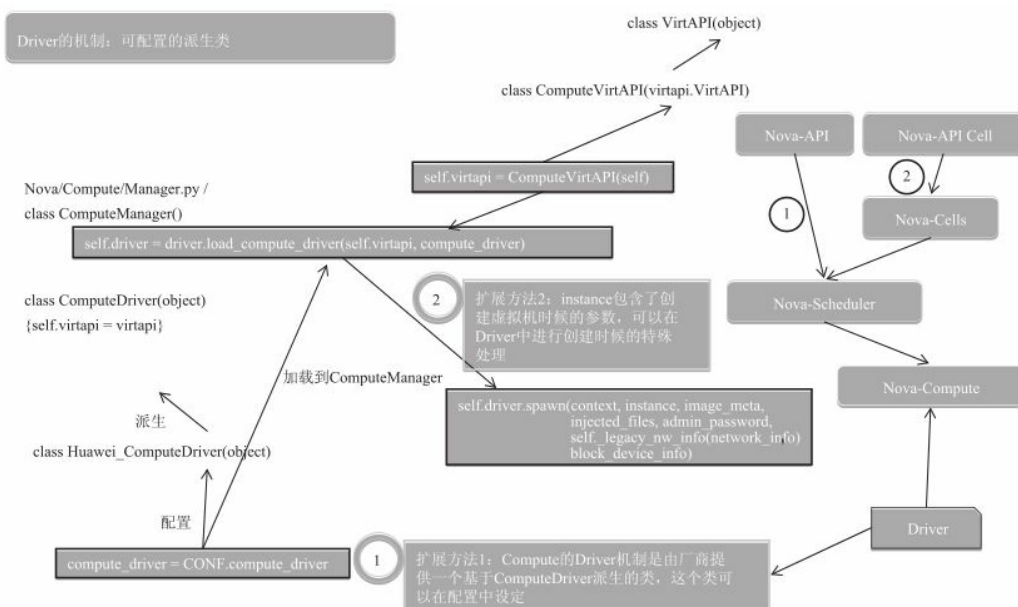


图3-26 Nova Driver的机制

除了上面提到的driver/plugin/静态配置/动态配置/参数自定义这些扩展方案，OpenStack还提供灵活的扩展机制实现API功能扩展（见表3-6）。

表3-6 OpenStack的扩展机制

术语（API扩展机制的术语）	含义	举例
资源（Resource）	Restful接口处理的对象	Server、Host Aggreagte就是资源
派发路由（Route）	Restful请求映射到其处理所在的类和方	因为对象之间存在关系，比如TenantID/Server/Metadata，到Metadata需要设置父对象是Server/，而Server/的父对象是TenantID
控制器（Controller）	Controller是指处理Resource的类	AggregateController就是处理Host Aggregate的类

如图3-27所示，只需要增加文件和修改配置，不需要修改源代码，就可以扩展资源及其API，非常灵活和方便。

在Nova内对计算资源域进行划分，一个Region是一个OpenStack实例，有独立的API服务（见图3-28）。在一个OpenStack实例下，可以包含多个Availability Zone，每个Availability Zone是一个故障域，比如共享电源的一个机房。在Availability Zone内可以对主机进行Host Aggregate和Group划分，比如一个机柜，有A厂家服务器和B厂家服务器，它们共享一个存储，可以作为一个Host Aggregate，但是要作为两个Group。对于API来说，Availability Zone是可见的资源区域调度调度单位，而Host Aggregate和Group这两个物理主机的资源分类概念可以为OpenStack带来灵活调度机制，满足资源多样化的需求。

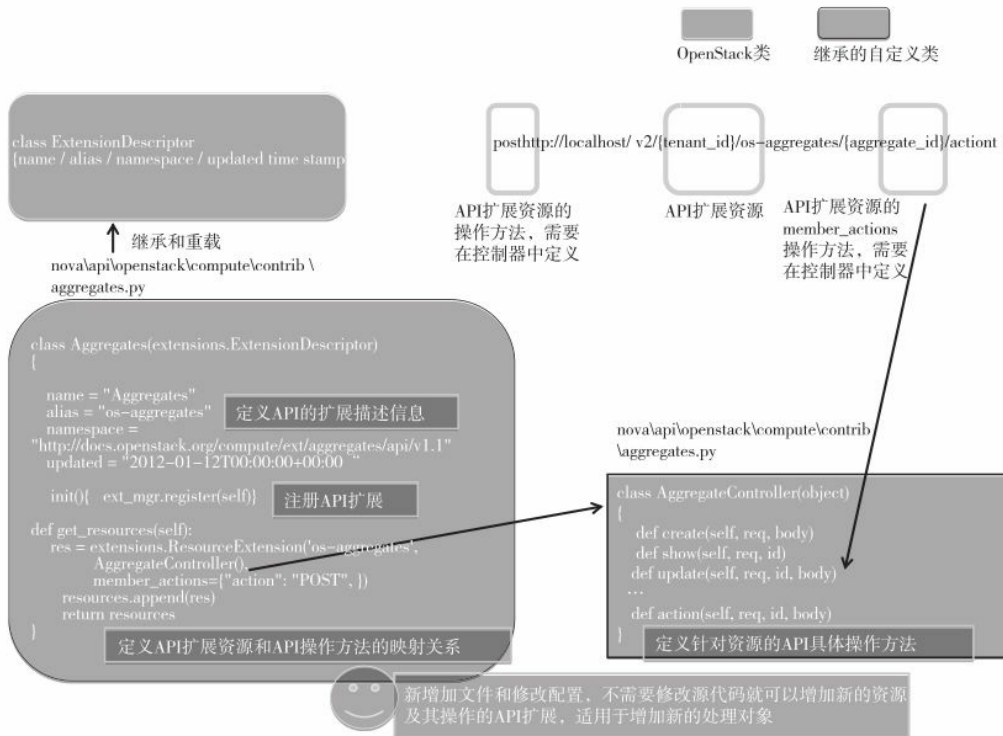


图3-27 增加资源及其API扩展方法

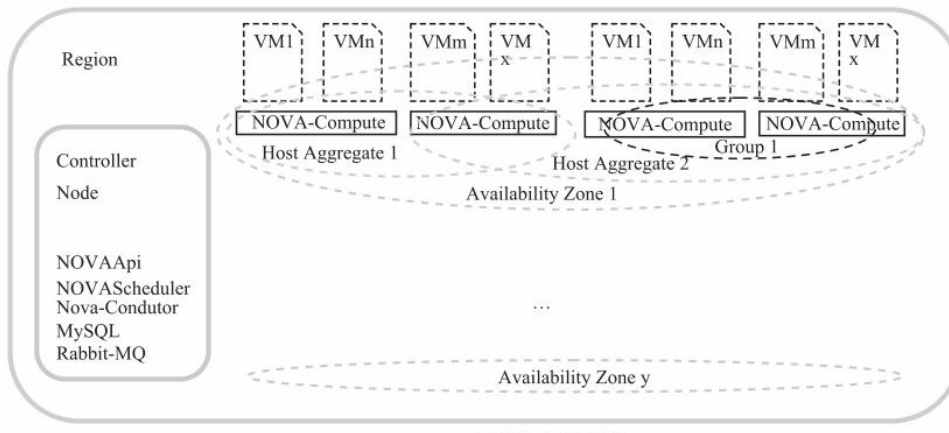


图3-28 Nova计算资源域划分

Nova还提供一种大规模资源管理的机制，即Cell机制。API服务只有一个节点，每一个Nova-Cells可以有很多的子Nova-Cells。调度时，先选定Cell，然后再在Cell那里进行Nova-Scheduler的调度。一个Cell可以是一个数据中心，也可以是一个机房。目前Cell机制还在不断优化中（见图3-29）。

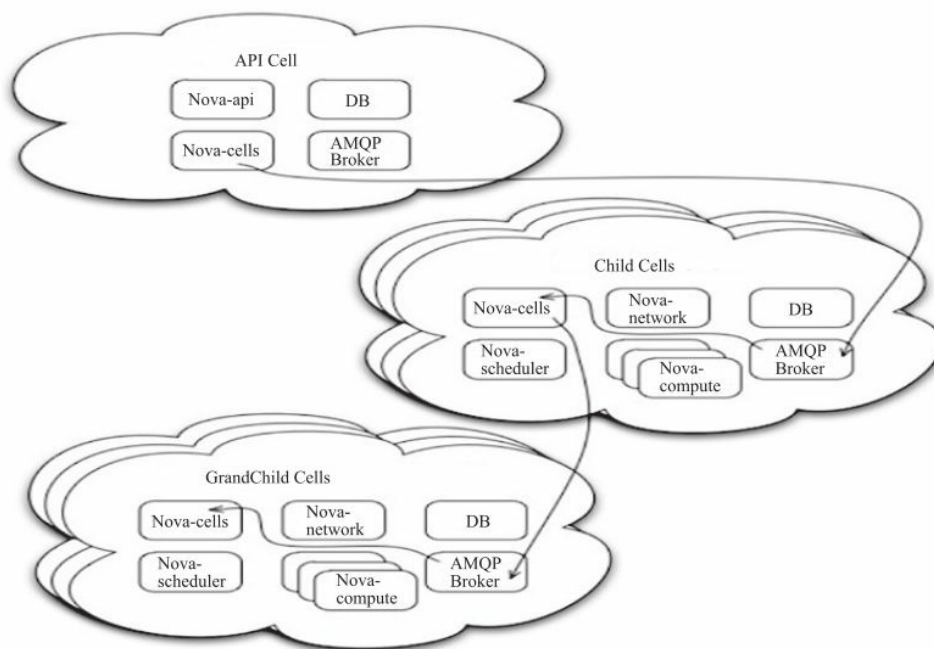


图3-29 Nova Cells机制

3.3.3 OpenStack的块存储服务：Cinder

通过Driver机制，Cinder可以支持不同类型存储，以提供块存储服务（见图3-30）。

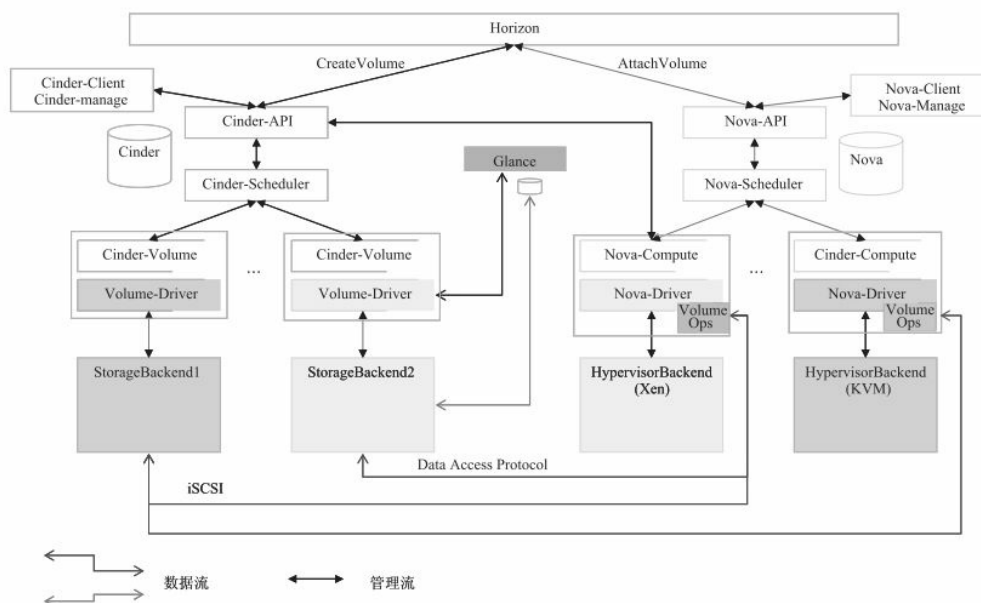


图3-30 Cinder的架构

目前支持的后端系统非常多，比如：

(1) iSCSI类后端系统

- IET+LVM/TGT+LVM
- Solaris
- HP Lefthand
- IBM XIV/StorwizeSVC
- Nexenta
- Dell EqualLogic
- SolidFire
- NetApp
- HUAWEI

(2) NFS (NAS) 类后端系统

- NetApp
- Glusterfs

(3) 其他类后端系统

- Ceph
- Sheepdog

Cinder提供了块存储设备的丰富功能，如卷的CRUD、卷挂载卸载、卷快照、从卷创建卷、从快照创建卷、从镜像创建卷、从卷创建镜像、卷

扩容、卷迁移等功能。

Cinder通过类似Nova的API、Scheduler、Cinder Volume的架构，支持系统的水平扩展能力（见图3-31）。

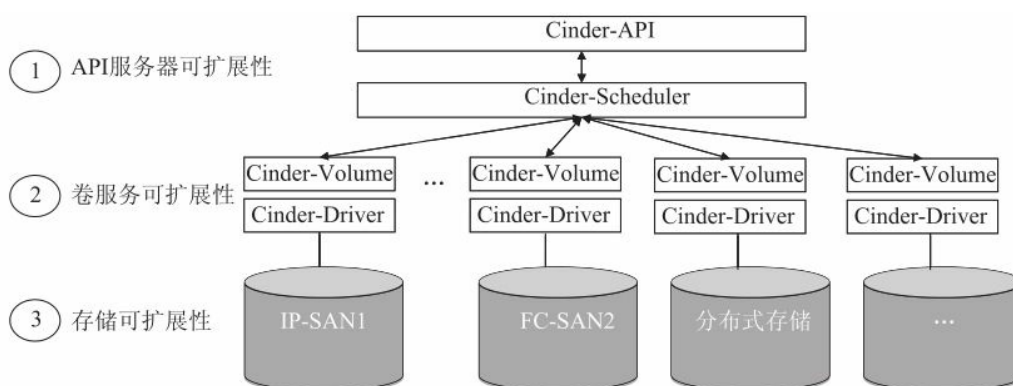


图3-31 Cinder的块存储服务可扩展能力

3.3.4 OpenStack的网络服务：Neutron

Neutron的核心是提供一个标准API集合下的Plugin机制。可以由各种网络厂商来实现具体的网络能力，也可以外接SDN Controller对网络实现更为智能的虚拟网络业务发放和流量控制（见图3-32）。

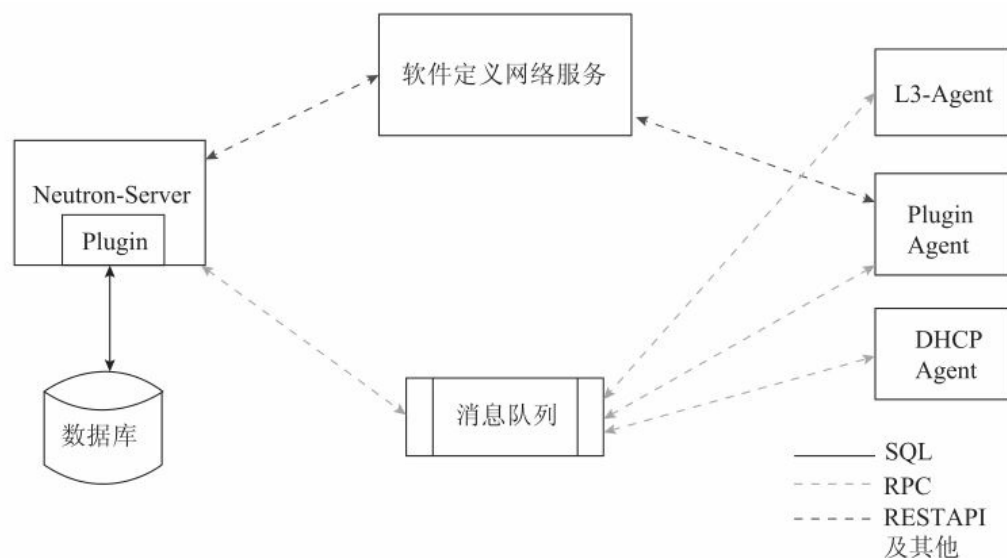


图3-32 Neutron的架构

Neutron的对象模型如图3-33所示，当前可以实现如下虚拟网络业务。

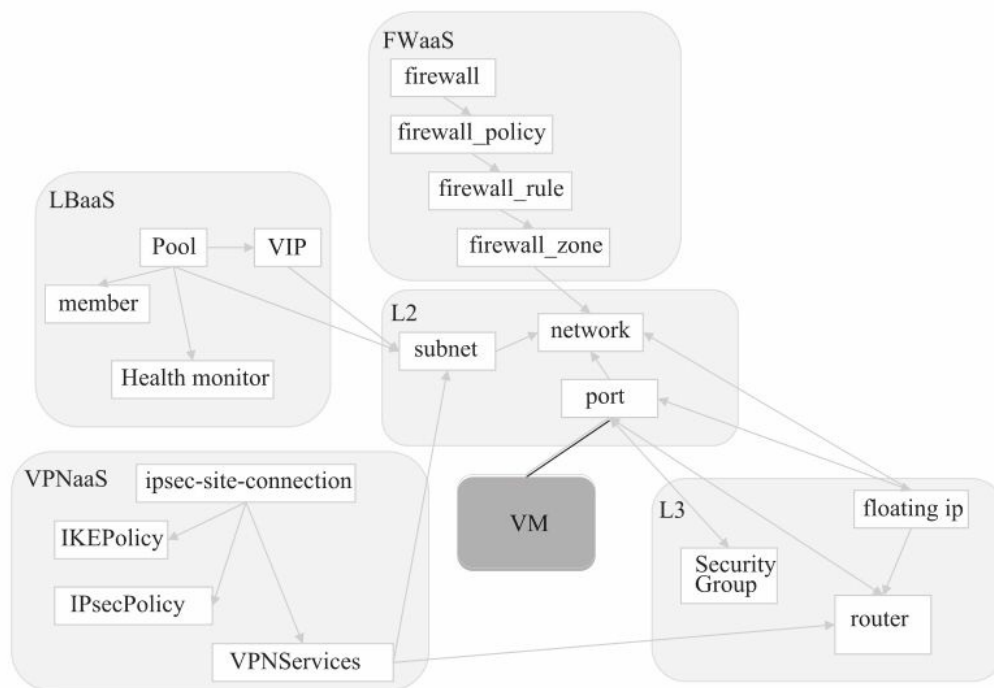


图3-33 Neutron的对象模型

OpenStack提供如下几种基本网络模型及其混合的网络模型（见图3-34、图3-35、图3-36）。

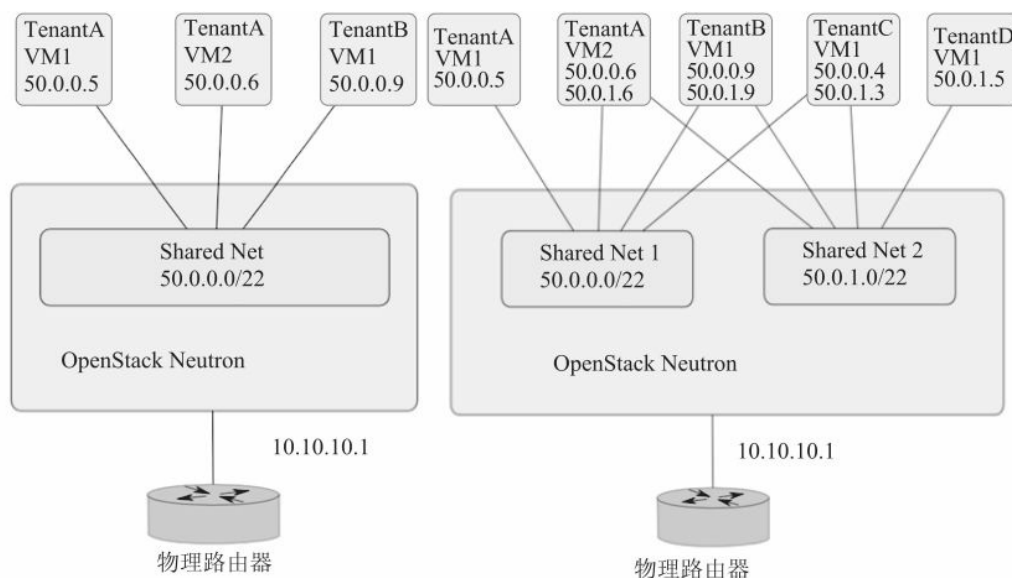


图3-34 Flat网络模式

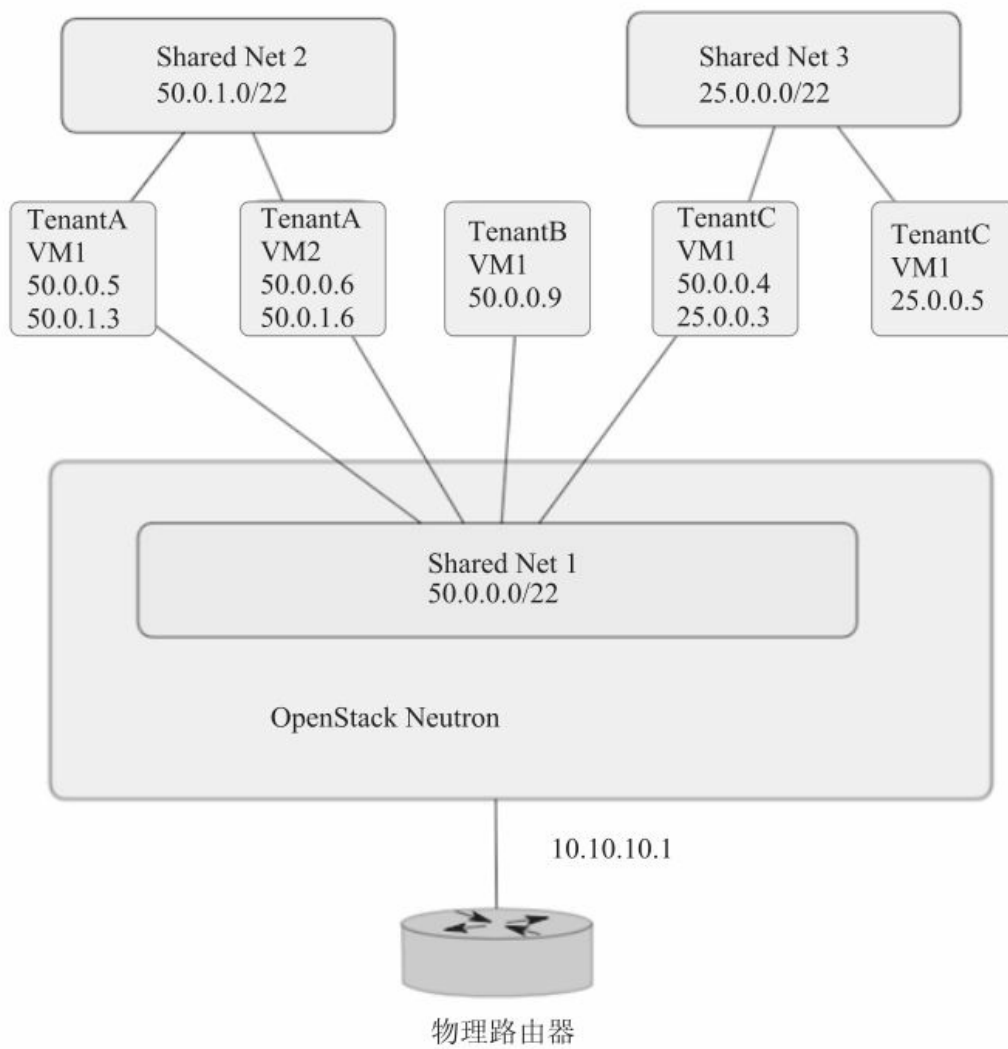


图3-35 混合网络模式

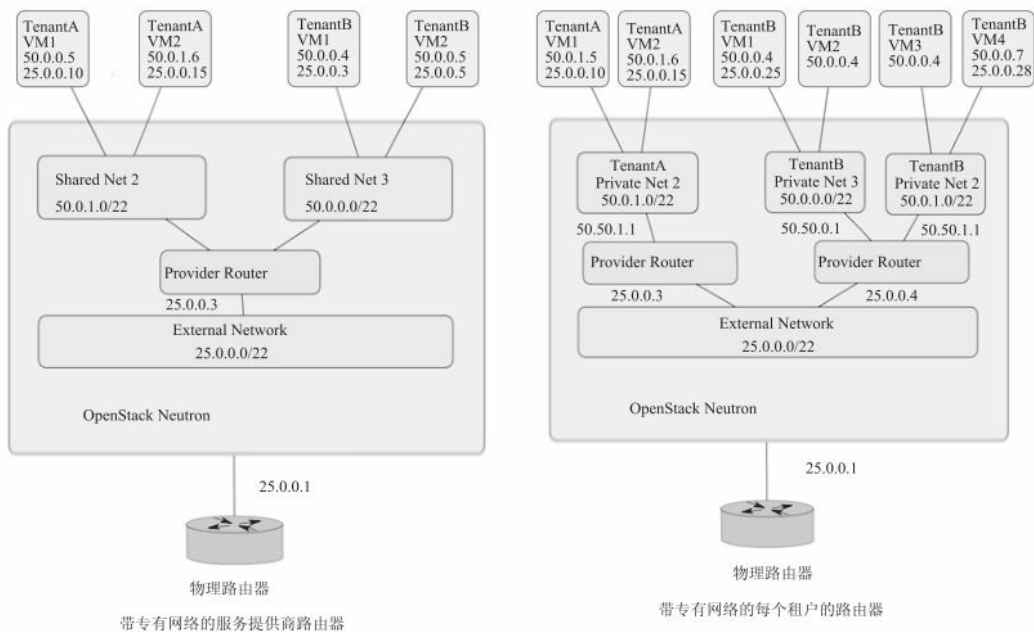


图3-36 Neutron共享Router模式和租户级Router模式

Neutron提供单Plugin和多Plugin模式，如图3-37所示。

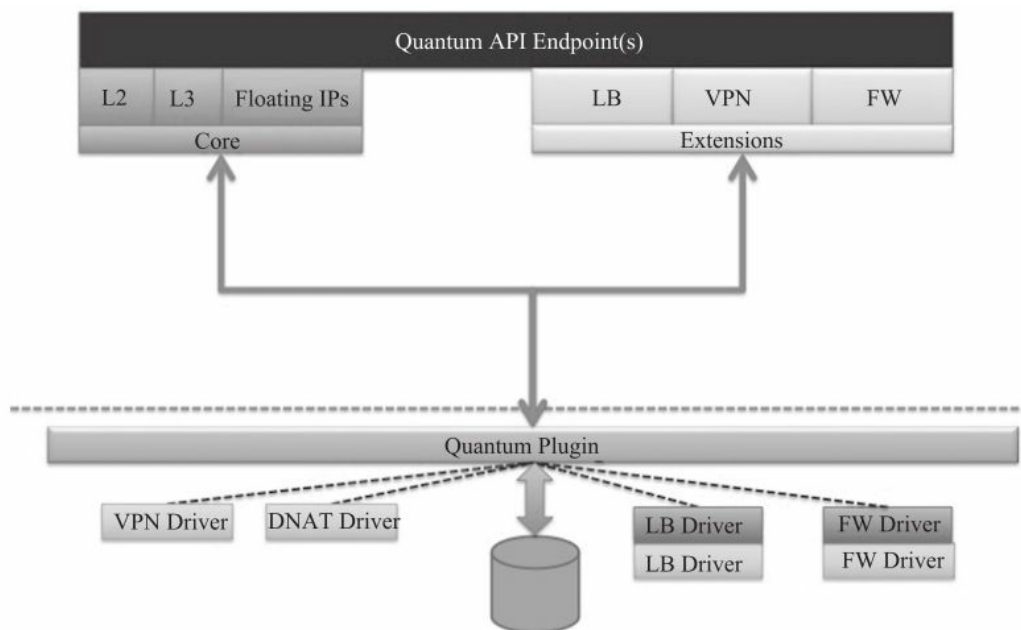


图3-37 Neutron单Plugin模式

单Plugin模式是一个Plugin实现所有的Neutron的功能，从L2到L3到扩展服务，如LB、FW、VPN等，如图3-38所示。

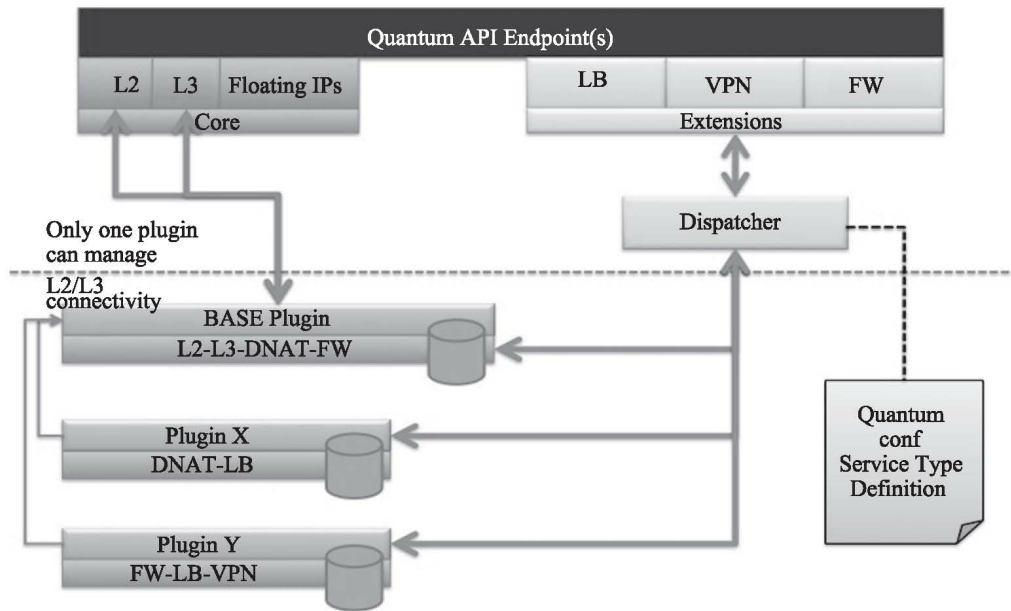


图3-38 Neutron多Plugin模式

多Plugin模式是一个Plugin只实现部分Neutron的服务能力，但是Grizzly版本只支持一个Base Plugin提供基本的L2/L3能力，扩展网络服务能力支持多Plugin。在Havana版本，L3能力也被作为网络扩展服务插件，L2则引入ML2插件，实现多种Layer2 Driver可以并存的机制（见图3-39）。

Neutron Server										
ML2 Plugin								API Extensions		
Type Manager			Mechanism Manager							
GRE TypeDriver	VLAN TypeDriver	VXLAN TypeDriver	Arista	Cisco Nexus	Hyper-V	L2 Population	Linuxbridge	Open vSwitch	Tail-F NCS	...

图3-39 Neutron的ML2机制

Neutron社区也在不断发展中，由于Plugin模式使得多厂家在单个生产环境中共存比较困难，因此都在走类似ML2的多种异构网络基础设施可以并存的Driver、Agent模式。

我们再以Neutron（原Quantum）为例，可以看到Plugin机制实际上与Driver机制非常类似，Quantum Manager会去扫描是否有Plugin的继承类，如果有，则把这些Plugin加载到运行系统中。同样地，其不需要修改源代码就可以扩展OpenStack的能力（见图3-40）。

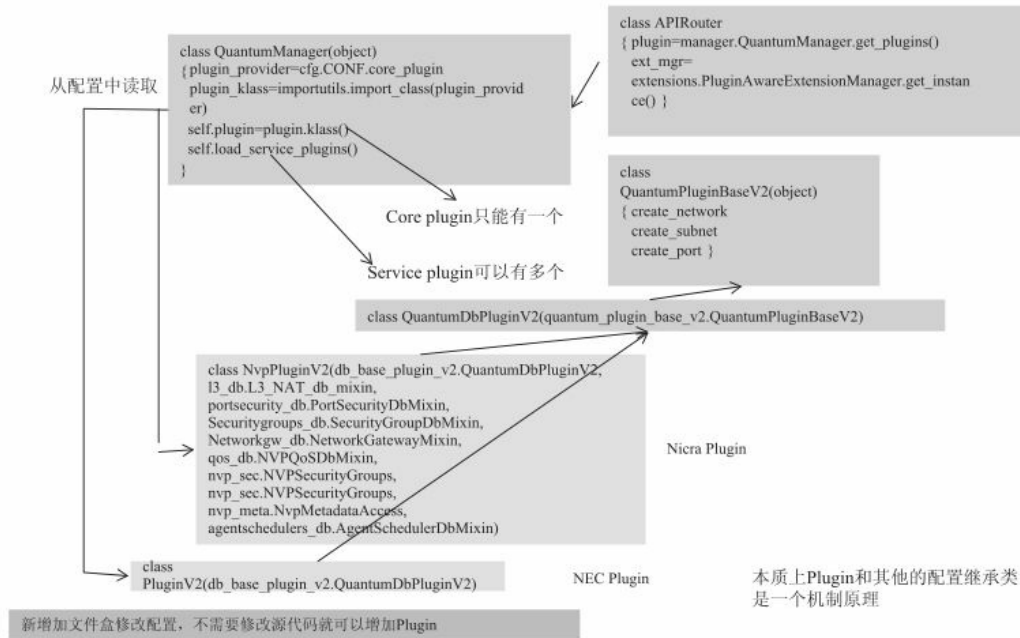


图3-40 Neutron的Plugin机制

3.3.5 OpenStack的镜像服务：Glance

镜像服务为虚拟机的创建提供了基本的镜像。Nova在创建虚拟机的时候会从Glance下载镜像。创建虚拟机之前，一般系统管理员会上载一些全局性的系统镜像，租户也可以对自己运行中的VM创建快照，快照会作为镜像上传到Glance。当然租户也可以在其他地方先做好镜像，再上传到Glance中。

Glance作为OpenStack的镜像服务，通过Driver机制，支持多种镜像后端存储，如Swift、S3、文件系统等（见图3-41）。

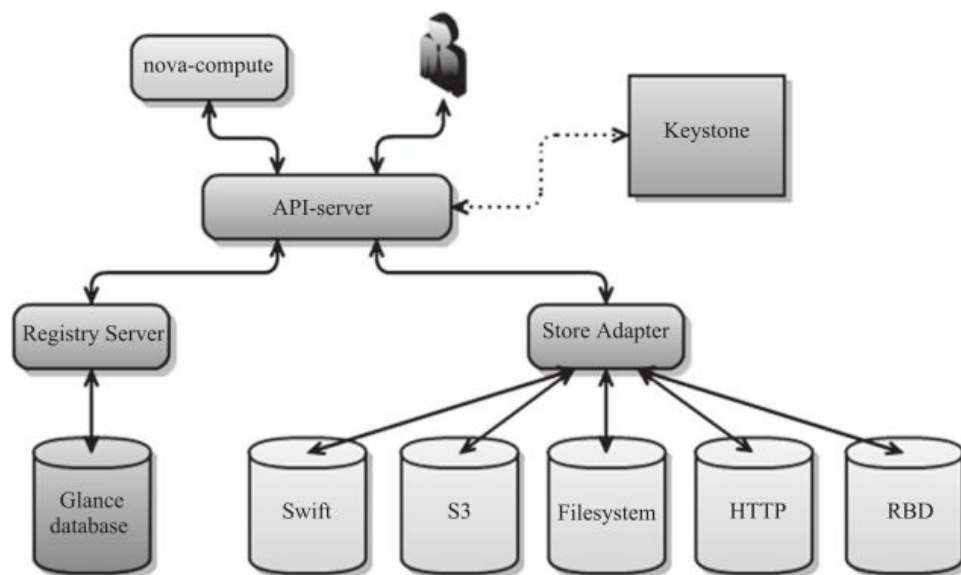


图3-41 Glance的可插拔可扩展架构

Glance支持如下镜像、模板格式：

- Raw;
- Machine (kernel/ramdisk outside of image, a.k.a. AMI/AKI/ARI);
- VHD (Hyper-V);
- VDI (VirtualBox);
- qcow2 (Qemu/KVM);
- VMDK (VMWare);
- OVF (VMWare, 其他);
- OVA;
- AMI。

3.3.6 OpenStack的身份服务：KeyStone

OpenStack的KeyStone支持集成不同的后端系统（如LDAP、SQL Server等关系型数据库，KVS.....）提供基于角色的用户身份管理，进行鉴权和认证管理。

KeyStone提供两种令牌（Token）供OpenStack的API安全访问，一种是UUID类的Token，在用户通过鉴权后，临时生产一个UUID，NOVA/Cinder/Neutron等服务在接受到API请求后，需要到KeyStone中去检查这个UUID是否合法，如果合法，则允许API访问。在大规模的云中，KeyStone会成为性能的瓶颈，因为所有API访问都要用到KeyStone。

因此社区从性能考虑，引入PKI的Token，相对于UUID Token，其好处是Nova/Cinder/Neutron等服务通过预先取好的证书对Token进行认证，减少了KeyStone的压力，缺点是Token会比较大，特别是在多Region下租户的访问权限和范围很大的时候，Token本身就是一个负担（见图3-42）。

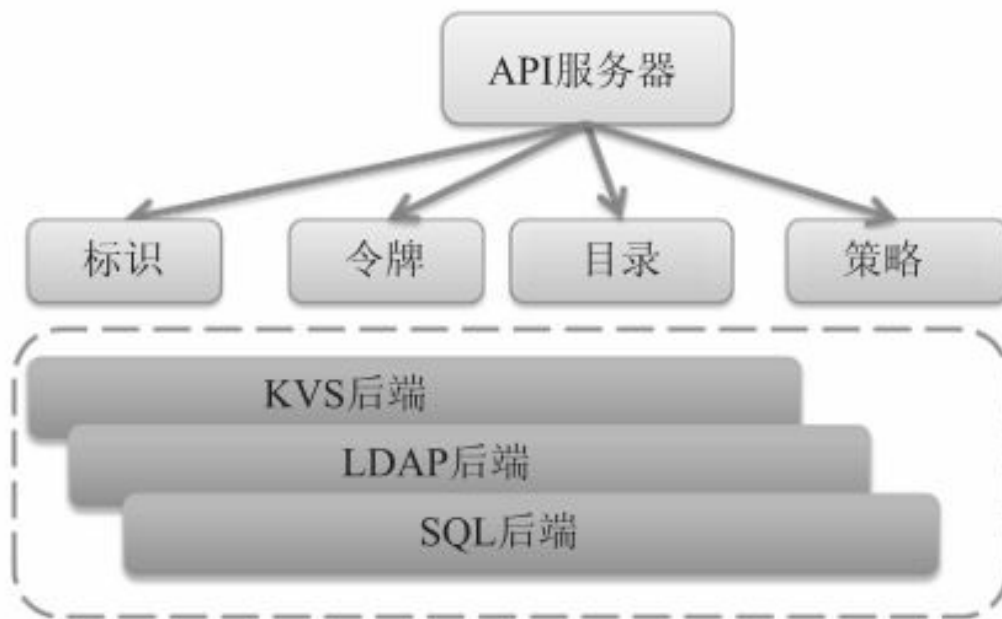


图3-42 KeyStone可扩展可插拔的灵活架构

KeyStone是实施OpenStack生产系统时非常关键的一个服务，涉及对

OpenStack云的管理模型。

Token和Credential是一个用户（User）访问云时需要获得的令牌或者密令。Token和Credential都是在针对特定的服务Service（如Nova、Cinder等）和访问地址endpoint（http://192.168.0.10/Nova/V2）时有效。

Project（OpenStack有时也用Tennat这个词）是资源的集合，资源会散布在Service/endpoint中。用户（User）要访问Project的资源，则需要在Project中具有一个角色（Role），并且通过Policy来控制角色对Project资源访问的权限。为了方便设置权限，可以把一组用户放在Group中。而Domain就是Project/User/Role/Group的集合。一个Domain可以有多个Project、多个User、多个不同的Role，多个Group（见图3-43）。

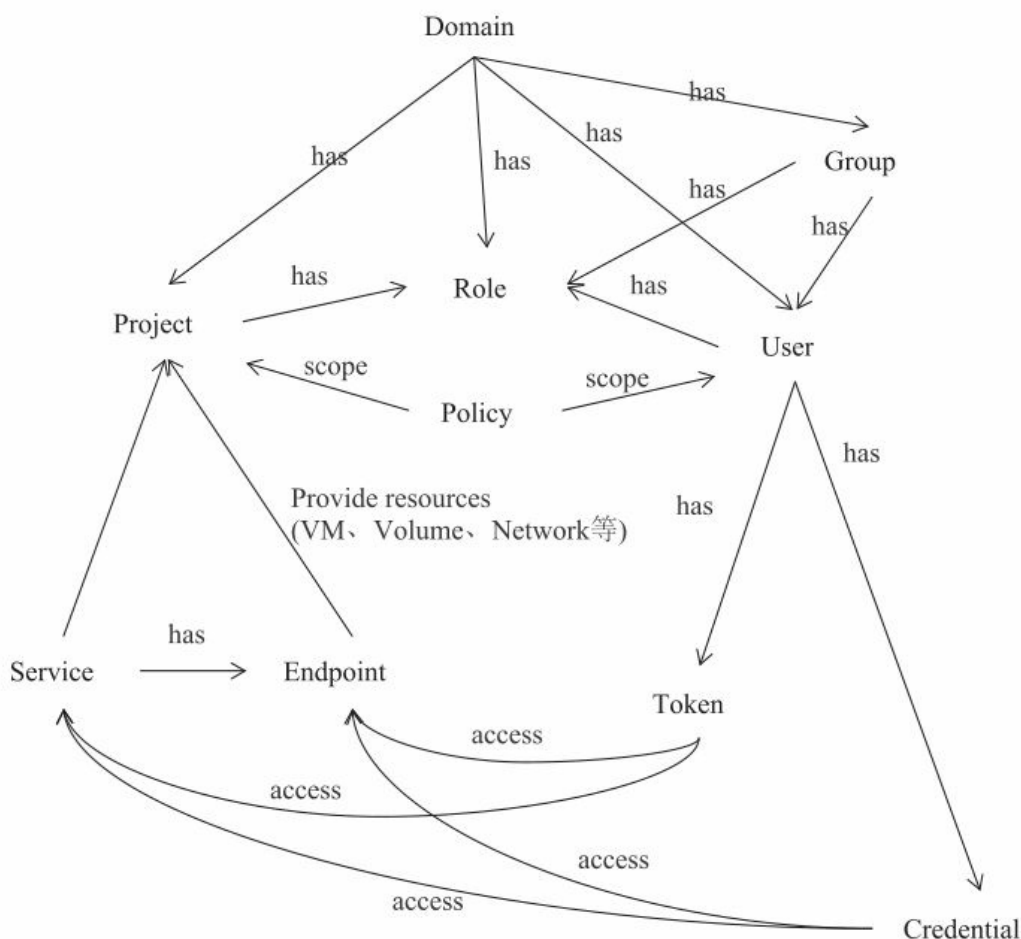


图3-43 Keystone的对象关系

3.4 开源和社区发展

3.4.1 Hypervisor社区发展

1. 社区发展

Xen和KVM是开源虚拟化技术的代表，都是由开源社区开发的，KVM的部分代码也是直接从Xen中移植过来的，KVM的很多开发者也都来自Xen项目组。KVM和Xen是两种不同的虚拟机实现方式，各有不同的优点和使用场合，这其中有来自社区人员的不懈努力，背后也有着众多企业的支持，这充分体现了开源开发模式的优越性和先进性，可以说Xen和KVM是这种社区开发模式的结晶。

图3-44展示了Xen和KVM的发展历史。

年份	Xen	KVM
1999	规划Xenserver	
2001	0.x，内部改造Nemesis微内核	
2002	1.x，内部开发，硬件驱动在Xen中	
2003		
2004	2.x，XenSource成立	
2005	3.0支持HVM，x86_64，vSMP	
2006		KVM发布
2007	Citrix收购XenSource、XenServer，建立xen.org	合入内核2.6.20
2008		Red Hat收购
2009	XCP(Xenserver社区版)	RHEL5.4支持KVM(同时支持Xen)
2010	Dom0代码合入Linux内核3.0	RHEL6.0仅支持KVM(RHEV2.0)
2011		RHEV3.0发布，Ovirt发布
2012		Red Hat加入OpenStack联盟
2013	Citrix把Xen捐给Linux Foundation	

图3-44 Xen、KVM历史发展进程

Xen虚拟化发展

Xen最初是作为剑桥大学的一个项目，目前由XEN.ORG社区负责它的开发及维护，已经在开源社区中得到了极大的发展。

（1）Xen的起源：2000年左右，Xen起源于英国的剑桥大学计算机实验

室，该实验室开发了Xen开源项目。这个项目包括虚拟机监控器，即Xen环境的核心组成部分。除了剑桥大学，IBM、AMD、HP、Red Hat和Novell都参与了Xen开源项目。由于Xen方法使虚拟化领域迈出了一大步，因此Xen的创始人成立了他们自己的公司XenSource。

(2) XenSource被收购：Citrix在2007年以5亿美元的价格收购了XenSource公司，因此Citrix有了自己的hypervisor产品Xenserver，在Linux服务器领域，Xen似乎成为了VMware之外的最佳虚拟化选择。

(3) 合入Linux Kernel 3.0：作为一项Linux平台上的虚拟化技术，Xen在很长一段时间内没有被接受到Linux内核的代码当中，直到在2011年6月发布的Linux内核3.0中才加入了对Xen的支持。

(4) 纳入Linux Foundation：2013年，Citrix和Linux基金会联合宣布，Xen虚拟化平台的开源社区活动将纳入Linux基金会合作项目。Linux基金会将利用其成熟的合作开发模式，为全新的Xen Project计划提供基础设施、开发指导和协作网络。

KVM虚拟化发展

KVM是目前最火热的开源虚拟化解决方案，由爱尔兰的公司Qumranet发起，并于2006年8月推向社区，2008年被Red Hat收购，成为Red Hat的主推虚拟化计划。

(1) KVM起源

KVM虚拟机最初是由一个以色列的创业公司Qumranet开发的，作为他们的VDI产品的虚拟机。为了简化开发，KVM的开发人员并没有选择从底层开始新写一个Hypervisor，而是基于Linux Kernel，通过加载新的模块，使Linux Kernel本身变成一个Hypervisor。2006年10月，在先后完成了基本功能、动态迁移以及主要的性能优化之后，Qumranet正式对外宣布了KVM的诞生。

(2) 进入Linux Kernel

2006年10月，KVM模块的源代码被正式接纳进入Linux Kernel，成为内核源代码的一部分。KVM最早由Avi Kivity等人开发，并于2006年8月推向社区，10月被Linux社区接受。它以其代码简单、易于理解掌握以及不需要重新安装等优点很快受到了业界的欢迎及Linux项目创始人

Torvalds的支持，并于2006年底被集成进Linux 2.6.20版内核，从此成为Linux的一个组成部分。

（3）被Red Hat收购

2008年9月4日，Red Hat公司出资1.07亿美元，收购了Qumranet，从而成为了KVM开源项目的新东家。正是因为此次收购，Red Hat公司有了自己的虚拟化解决方案。

（4）RHEL5.4支持KVM

2010年11月，Red Hat公司推出了新的企业版Linux-RHEL 6，在这个发行版中集成了最新的KVM虚拟机，而去掉了在RHEL 5.x系列中集成的Xen。

（5）RHEV3.0发布，Ovirt发布

2011年11月，Red Hat、IBM、英特尔、思科、Canonical、NetApp与SUSE赞助成立Ovirt社区。其目的是创建一个有活力的开源社区，涵盖虚拟化管理堆栈各个层级，包括Hypervisor、管理、图形用户界面、API等，提供一个功能丰富的服务器虚拟化管理系统。Red Hat向Ovirt社区贡献了自己的虚拟化管理软件，IBM则捐出了Memory Overcommit Manager。如今，Ovirt已是一个全面的生态系统。

2. 商用情况

如今，企业正在部署或已经实施虚拟化技术，以便提高IT基础架构的利用率和经济性。利用数据中心虚拟化，这些企业推动了关键应用程序的高可用性与快速恢复。最近几年，VMware、思杰和微软等虚拟化技术占据了数据中心大部分领地。但是，商用的解决方案不仅部署与运维成本昂贵，而且容易被厂商锁定。企业用户希望保持对IT的控制能力，确保底层虚拟平台具有开放性，并有一个强大的生态系统对其进行支撑。

Xen技术商用情况

经过10年的发展，Xen技术已拥有超过1 000万用户，并吸引了来自众多机构的参与和贡献，包括亚马逊、AMD、剑桥大学、思杰、富士通、英特尔、美国国家安全局（NSA）、甲骨文、华为和SUSE等。随着云时代的到来，一些技术领先的企业组织还在继续推动Xen Project在云领

域的发展，其中包括亚马逊Web服务（AWS）、AMD、Bromium、Calxeda、CA、思科、思杰、谷歌、英特尔、甲骨文、三星和Verizon。

KVM技术商用情况

2011年初，IBM和Red Hat，联合惠普和英特尔一起，成立了开放虚拟化联盟（Open Virtualization Alliance），一起声明要加速KVM投入市场的速度，由此避免VMware一家独大的情况出现。联盟成立之时，Red Hat的发言人表示：“大家都希望除VMware之外还有一种开源选择。未来的云基础设施一定会基于开源.....我们想要营造一个小厂商们可以轻松加入的生态环境。”

于是，开放虚拟化联盟红红火火地成立了。从2011年5月到8月这短短3个月间，开放虚拟化联盟的成员已经增加到将近300个，联盟发展的速度十分可观。IBM现在全线硬件都对Red Hat Linux和KVM进行了大量的优化，有60多名开发者专门开发KVM相关的代码。

3.4.2 Cloud OS社区发展

1. 社区发展

由于OpenStack的开放性，IT各领域排名前三位的厂商都支持OpenStack，并深度参与，提供OpenStack相关的集成解决方案和服务。

与OpenStack相比，CloudStack主要由原Citrix工程师主导，由于架构上的原因，CloudStack可供扩展的地方不多（重构优化、插件、文档），一部分插件已由Citrix提供，只有少部分由第三方/服务商提供：

- Caringo Contributes Object Store Plugin
- Nicira Controller by Hugo
- Ceph/RBD Support by Wido
- CLVM for KVM by Marcus
- CloudEra (Hadoop Backed Object Store)

- Midokura (SDN Controller)
- Basho-Object Store

总体上，OpenStack与CloudStack的比较如下。

从两个开源社区的发展速度看，CloudStack生态链上参与厂商较少，OpenStack在参与度、活跃度、厂商贡献积极性等各方面呈现出远超CloudStack的发展势头。

CloudStack在局部特性上（GUI、安装部署、资源管理）成熟度较强。在业务特性上（如计算资源池、块存储等），OpenStack与CloudStack能力相当。CloudStack缺少对象存储，OpenStack的对象存储（Swift）功能卓越且广泛应用。在网络特性上，CloudStack成熟，OpenStack网络模型灵活且发展迅猛，但有待成熟。

OpenStack系统架构在开放性、扩展性、分层解耦、对外服务API等方面要远强于CloudStack，更适合灵活部署大规模云服务基础平台。

在管理与自动化方面，CloudStack具备较成熟的封闭管理系统，OpenStack具备较强的开放性，由用户与系统集成商选择业界最优方案。

由于软件架构设计的原因，OpenStack呈现出更为强劲的生态系统，后劲更足；CloudStack则表现为当前商用成熟度更高，将带动社区继续往前发展。

2. 商用情况

在亚特兰大2014年的峰会上，OpenStack基金会公布了最新的OpenStack部署情况：美国最多，其次是中国。调查问卷共反馈了506家OpenStack部署（见图3-45）。

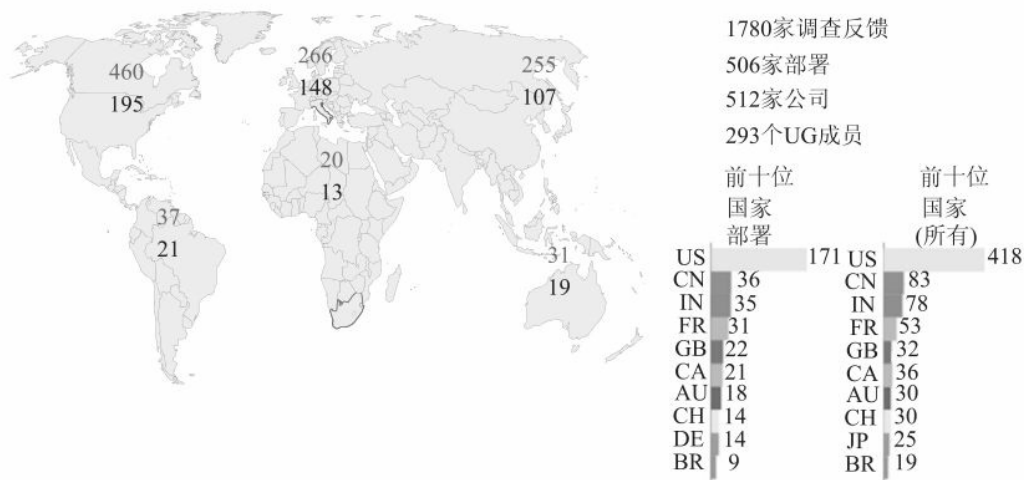


图3-45 调查问卷反馈

部署的行业分布如下，其中IT行业还是最多，占58%，电信行业有6%。随着NFV组织和OpenStack社区的互动，NFV已经基本倾向于用OpenStack来部署其云基础设施，相信电信行业的OpenStack交付部署会越来越多（见图3-46）。

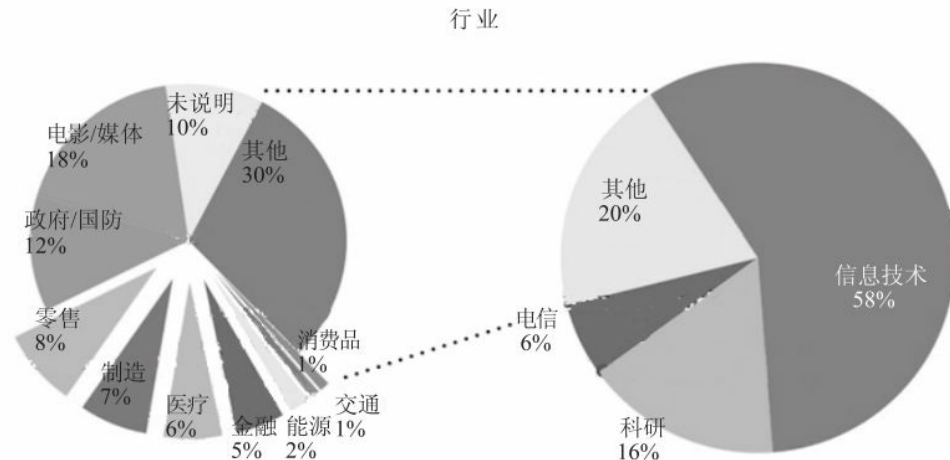


图3-46 OpenStack部署的行业分布

3.5 开源还是闭源

早期的开源，注重的是开放源代码；近期和未来的开源发展，更加注重的是一个开放的架构，所开放的源代码也是为这个开放的架构服务；再说得高级一些，开源最终是在朝着开放的生态链方向发展。在这个开放

的生态链中，大家（包括厂商、用户、研究机构、个人开发者）可以自由选择（来自不同厂家的体系模块），自由发展（做符合自身的定制开发），自由竞争（不同厂家可以开发相同的模块组件，比拼性能）。完全自由虽然是一个理想，但至少在向这个方向努力。

支撑开源健康发展的不只是开源的精神，更是开源的商业模式。因为“精神”虽然很重要，但不能当饭吃，先让生态链中的每个环节、每个成员吃饱饭，才能考虑精神享受，对于早期为开源做出重大贡献的大牛们也是如此。在那个IT人才稀缺的年代，衣食无忧之余才会把源代码公开的。“免费”是开源的一大特色，但不能理解成“彻底与完全”的免费，而应该理解成“最优的成本构成”（基于开源的产品竞争会更加透明化，对用户绑定概率低，性价比更加合理），开源用户需要一定的付出（可能是购买基于开源的产品、咨询与技术支持服务的费用；可能是申请技术认证的费用；可能是对开源社区的直接赞助费用；还可能是贡献自己的代码、数据，等等），才会得到满意的回报。

闭源不会因为开源的蓬勃发展而快速消失，但闭源与开源的产品及系统模块之间的配合会更加紧密。闭源也会更多地融入开源逐步主导的开放产业链之中。在开放的架构中，闭源模块和开源模块之间的竞争会更加激烈，这对用户来说是一件大好事，因为即使闭源产品继续存在，但出于竞争的需要，其也会为用户提供更加实惠合理的价格。

对于科研机构和处于学习阶段的个人开发者而言，丰富的开源软件无疑是这个时代的福音。其不仅可以免费使用，降低科研和学习成本，还可以进行任意的试验和修改，是取得科研成就和提升个人软件开发水平的极佳工具。但是如果希望科研成果或者个人的综合技能接上地气，即实现商业价值转换，开源软件能帮的忙是很有限的。因为商业产品所看重的是稳定、高效、易用以及长期稳定的技术支持服务，这所需要的最大投入不是智力，而是体力。

对于IT系统供应商，基于开源软件开发的产品，能够帮助其快速完成产品化，缩短产品开发周期，快速推向市场。但正如前文所提及的，同样基于开源软件的产品众多，产品的同质化严重，会带来非常激烈的市场竞争，大家比拼的不仅包括价格，还有产品稳定性、易用性，更多的是长期、稳定且深入的技术服务能力。与科研机构比，企业有资金及人力优势。但企业和企业比，大家最终拼的几乎还是研发与服务的资金、人员的投入以及成本控制能力。

对于大多数开源软件的最终企业级用户而言，最大的价值不是免费，而是开源带来开放的架构。用户在开放的架构下，对架构中的模块可以有更多的供应商进行选择，而且模块可以更加容易替换，特别是在云的环境下，整个虚拟化平台都可以替换。这样可以避免被某一家IT供应商深度绑定（被深度绑定不仅会带来高昂的成本，而且用户期望的任何改动和服务支持都得看供应商的脸色，遇上傲慢的IT供应商，有时给钱都没用，让企业用户丧失业务灵活性）。当然，企业级用户采用基于开源产品的前提是，基于开源的产品综合效能与可用性可以进行业务商用。企业用户在使用基于开源产品时，往往容易产生一个误区，那就是期望基于开源的产品在性能、成本、可用性、易用性、服务支持能力等方面能够全面超越闭源产品。这个理想虽然是好的，但不是很现实，这里有一个很简单的逻辑，开源产品面向闭源厂商也是开放的，只要不是很傲慢的闭源厂商，就可以时刻学习开源产品的最新特性为自身所用，却不把自己优化的内容开放给别人。我们指望完全的开源全面超越闭源，可能性不是很大。所以企业是否选择开源的决策，还是要回到对开源核心价值的判断上，那就是企业是否觉得有必要选择一个“深入开放”的IT架构。

在企业级用户里，还有一部分以大型互联网企业为代表的，拥有强大研发能力的特殊用户，他们的IT核心平台基本都是基于开源软件搭建的，而且是基于开源自主开发的。开源对他们的最大价值，是面向自身业务的深入定制能力。大型互联网企业的IT平台规模庞大，业务特性独特，传统闭源产品的推出往往为了满足广大普通企业用户的需求，对于大型互联网企业的独特需求，难于满足，而且因为同类需求客户数量少，闭源厂商更不乐意定制修改。为了克服这个问题，大型互联网企业为了自身业务发展，只能基于开源，针对自身业务进行深入的定制开发。这种定制开发，可以说投入巨大，而且涉及定制的层面越深，花销越大，还会带来平台稳定性、数据安全性等方面的运营风险。但从业务回报的角度看，承担这样的投入和风险还是值得的。

第4章 面向电信及企业关键应用的计算虚拟化

虚拟化技术因为节省能源和提升服务器的使用效率等诸多因素而得到了广泛的认可。如今，许多公司使用虚拟技术作为建设云计算的基础技术，用以提高硬件资源的利用率，进行灾难恢复，提高办公自动化水平。

虚拟化技术经历了漫长的发展时期，早在20世纪70年代，大型计算机就一直在同时运行多个操作系统实例，每个实例彼此独立，这时候虚拟化技术还没有得到广泛的普及。直到当今，软硬件方面的技术取得了巨大的进步，使得虚拟化技术在基于行业标准的大众化x86服务器上得以支撑电信和企业关键应用的部署，虚拟化技术逐步得到了广泛的认可，从而得以普及。

电信领域通常所说的应用，主要是组成电信数据网络和信令网络的网元中的逻辑功能实体，例如：无线网络中的基站控制单元、分组数据交换单元；移动通信网络中的IP多媒体域控制网元、移动交换控制网元；固定通信领域中的电话交换网元，以及支撑电信网络安全稳定可靠运行的运维管理系统，支撑电信业务运营的业务支撑系统等。

企业关键应用，主要是指与企业的日常运营密切相关的IT生产系统或者IT支撑系统所运行的各种软件业务应用程序。企业关键应用与国民经济中的各行各业的自身特点密切相关，所采用的软件技术分门别类、千差万别，具有非常大的差异性。

尽管电信和企业关键应用的种类丰富，但很多应用对云计算运行环境的要求比较雷同，归纳总结分为计算高性能低时延要求、可靠性要求、资源占用效率要求等。

本章针对上述关键性要求，选取了计算虚拟化中的关键技术予以阐述，来说明当前的技术发展现状。

4.1 计算虚拟化核心引擎：

Hypervisor介绍

虚拟化技术源于大型机，最早可以追溯到20世纪六七十年代大型机上的虚拟分区技术，即允许在一台主机上运行多个操作系统，让用户尽可能地利用昂贵的大型机资源。随着技术的发展和市场竞争的需要，虚拟化技术向小型机或UNIX服务器上移植，只是由于真正使用大型机和小型机的用户还是少数，加上各厂商产品和技术之间的不兼容，使得虚拟化技术不太被公众所关注。由于x86架构在设计之初并没有考虑支持虚拟化技术，它本身的结构和复杂性使得在其之上进行虚拟化非常困难，早期的x86架构并没有成为虚拟化技术的受益者。

20世纪90年代，虚拟化软件厂商采用一种软件解决方案，以软件模拟的方式使x86服务器平台实现虚拟化。对于这种纯软件的“全虚拟化”模式，每个Guest OS（客户操作系统）获得的关键平台资源都要由模拟层的软件来控制 and 分配，需要利用软件实现二进制转换，而二进制转换带来的开销使得“完全虚拟化”的性能大打折扣。

为解决性能问题，出现了一种新的虚拟化技术“半虚拟化”，即不需要二进制转换，而是通过对客户操作系统进行代码级修改，使定制的Guest OS获得额外的性能和高扩展性，但是修改Guest OS也带来了系统指令级的冲突及运行效率问题，需要投入大量优化的工作。

当前，虚拟化技术已经发展到了硬件支持的阶段，“芯片辅助虚拟化”技术就是把纯软件虚拟化技术的各项功能以硬件电路的形式来实现，可减少VMM运行的系统开销，同时满足CPU半虚拟化和二进制转换技术的需求，使VMM的设计得到简化，进而使VMM能够按通用标准进行编写。芯片辅助虚拟化技术除了在处理器上集成芯片辅助虚拟化指令，同时提供I/O方面的虚拟化支持，最终可实现整个平台的虚拟化。虚拟化技术的实现和发展向人们展示了虚拟化应用的广阔前景。

4.1.1 业界典型计算虚拟化架构说明

作为后续了解计算虚拟化关键技术的基础，本节简要说明一下计算虚拟化的架构，并介绍一些基础概念作为铺垫。

计算虚拟化技术的实现形式是在系统中加入一个虚拟化层，将下层的资源抽象成另一种形式的资源，提供给上层使用。

计算虚拟化技术的通用实现方案是将软件和硬件相互分离，在操作系统与硬件之间加入一个虚拟化软件层，通过空间上的分割、时间上的分时以及模拟，将服务器物理资源抽象成逻辑资源，向上层操作系统提供一个与它原先期待一致的服务器硬件环境，使得上层操作系统可以直接运行在虚拟环境上，并允许具有不同操作系统的多个虚拟机相互隔离，并发运行在同一台物理机上，从而提供更高的IT资源利用率和灵活性。

计算虚拟化的虚拟化软件层需要模拟出来的逻辑功能主要为高效、独立的虚拟计算机系统，我们称之为虚拟机，在虚拟机中运行的操作系统软件，我们称之为Guest OS。

计算虚拟化技术可以将单个CPU模拟为多个CPU，允许一个平台同时运行多个操作系统，并且应用程序可以在相互独立的空间内运行而互不影响。简单地说，计算虚拟化技术实现了计算单元的模拟和模拟出来的计算单元间的隔离。

虚拟化软件层模拟出来的每台虚拟机都是一个完整的系统，它具有处理器、内存、网络设备、存储设备和BIOS，因此虚拟机中运行的操作系统和应用程序与在物理服务器上运行的操作系统和应用程序并没有本质的区别（见图4-1）。

计算虚拟化的这个软件层，也就是虚拟机监控器（Virtual Machine Monitor, VMM），通常被称为Hypervisor。常见的Hypervisor软件栈架构方案分为两类，即Type-I型和Type-II型。

Type-I型（裸金属型）指VMM直接运行在裸机上，使用和管理底层的硬件资源，Guest OS对真实硬件资源的访问都要通过VMM来完成，作为底层硬件的直接操作者，VMM拥有硬件的驱动程序。

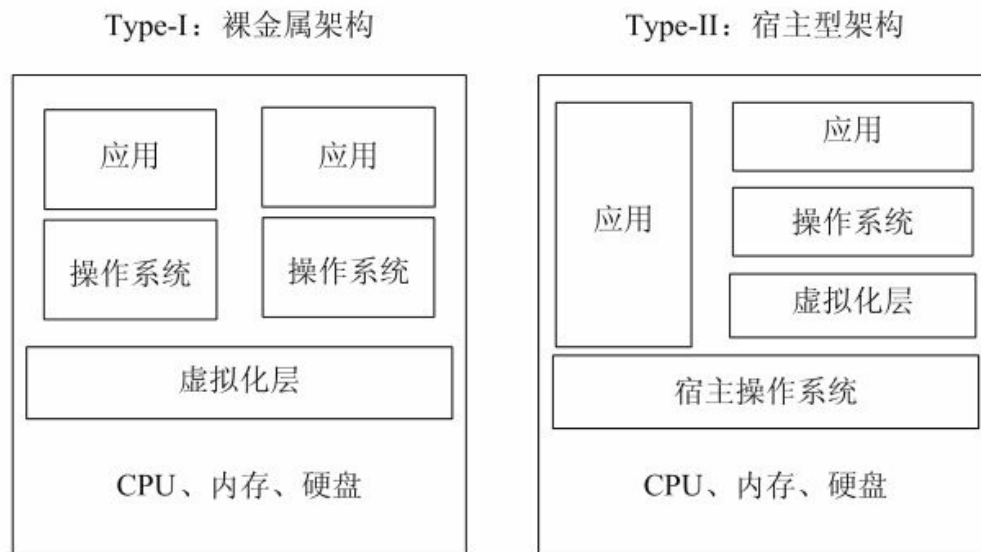


图4-1 虚拟化架构

Type-II型（宿主型）指VMM之下还有一层宿主操作系统，由于Guest OS对硬件的访问必须经过宿主操作系统，因而带来了额外的性能开销，但可充分利用宿主操作系统提供的设备驱动和底层服务来进行内存管理、进程调度和资源管理等。

我们进一步分析Hypervisor对于CPU指令的模拟和虚拟实例的隔离方式，计算虚拟化技术可以细分为如下几个子类。

全虚拟化（**Full Virtualization**）

全虚拟化是指虚拟机模拟了完整的底层硬件，包括处理器、物理内存、时钟、外设等，使得为原始硬件设计的操作系统或其他系统软件完全不做任何修改就可以在虚拟机中运行。操作系统与真实硬件之间的交互可以看成是通过一个预先规定的硬件接口进行的。全虚拟化VMM以完整模拟硬件的方式提供全部接口（同时还必须模拟特权指令的执行过程）。举例而言，x86体系结构中，对于操作系统切换进程页表的操作，真实硬件通过提供一个特权CR3寄存器来实现该接口，操作系统只需执行“`mov pgtable,%cr3`”汇编指令即可。全虚拟化VMM必须完整地模拟该接口执行的全过程。如果硬件不提供虚拟化的特殊支持，那么这个模拟过程将会十分复杂。一般而言，VMM必须运行在最高优先级来完全控制主机系统，而Guest OS需要降级运行，从而不能执行特权操作。当Guest OS执行前面的特权汇编指令时，主机系统会产生异常

（General Protection Exception），执行控制权将重新从Guest OS转到VMM手中。VMM事先分配一个变量作为影子CR3寄存器给Guest OS，将pgtable代表的客户机物理地址（Guest Physical Address）填入影子CR3寄存器，然后VMM需要将pgtable翻译成主机物理地址（Host Physical Address）并填入物理CR3寄存器，最后返回到Guest OS中。随后VMM还将处理复杂的Guest OS缺页异常（Page Fault）。比较著名的全虚拟化VMM有Microsoft Virtual PC、VMware Workstation、Sun Virtual Box、Parallels Desktop for Mac和QEMU。

超虚拟化（**Paravirtualization**）

这是一种修改Guest OS部分访问特权状态的代码以便直接与VMM交互的技术。在超虚拟化虚拟机中，部分硬件接口以软件的形式提供给客户机操作系统，这可以通过Hypercall（VMM提供给Guest OS直接调用，与系统调用类似）的方式来提供。例如，Guest OS把切换页表的代码修改为调用Hypercall来直接完成修改影子CR3寄存器和翻译地址的工作。由于不会产生额外的异常和模拟部分硬件执行流程，超虚拟化可以大幅度提高性能，比较著名的VMM有Denali、Xen。

硬件辅助虚拟化（**Hardware-Assisted Virtualization**）

硬件辅助虚拟化是指借助硬件（主要是主机处理器）的支持来实现高效的全虚拟化。例如有了Intel-VT技术的支持，Guest OS和VMM的执行环境自动地完全隔离开来，Guest OS有自己的“全套寄存器”，可以直接运行在最高级别。因此在上面的例子中，Guest OS能够执行修改页表的汇编指令。Intel-VT和AMD-V是目前x86体系结构上可用的两种硬件辅助虚拟化技术。

部分虚拟化（**Partial Virtualization**）

VMM只模拟部分底层硬件，因此客户机操作系统不做修改是无法在虚拟机中运行的，其他程序可能也需要进行修改。在历史上，部分虚拟化是通过全虚拟化道路上的重要里程碑，最早出现在第一代的分时系统CTSS和IBM M44/44X实验性的分页系统中。

操作系统级虚拟化（**Operating System Level Virtualization**）

在传统操作系统中，所有用户的进程本质上是在同一个操作系统的实例

中运行的，因此内核或应用程序的缺陷可能会影响其他进程。操作系统级虚拟化是一种在服务器操作系统中使用的轻量级的虚拟化技术，内核通过创建多个虚拟的操作系统实例（内核和库）来隔离不同的进程，不同实例中的进程完全不了解对方的存在。比较著名的虚拟化技术有Solaris Container、FreeBSD Jail和OpenVZ等。

4.1.2 满足电信和企业关键应用的计算虚拟化技术

如前文所述，为了满足电信和企业关键应用，计算虚拟化需要满足计算高性能低时延的要求、可靠性的要求、资源占用效率的要求。为了满足计算高性能低时延的要求，主要的关键性技术如下。

1. 精细化CPU调度技术

为了保证电信和关键企业应用运行的性能，就要求同一台物理机上的多个虚拟化运行实例所获取的资源既能满足其运行的需要，同时不互相产生干扰，精细化的CPU调度技术应运而生。

精细化CPU调度技术主要指的是CPU上下限配额及优先级调度技术。

现代计算机体系结构一般至少有两个特权级（即用户态和核心态）用来分隔系统软件和应用软件。那些只能在处理器的最高特权级（内核态）执行的指令称之为特权指令，一般可读写系统关键资源的指令（即敏感指令）绝大多数都是特权指令（x86存在若干敏感指令，这些指令是非特权指令）。如果执行特权指令时处理器的状态不在内核态，通常会引发一个异常，从而交由系统软件来处理这个非法访问（陷入）。经典的虚拟化方法就是使用“特权解除”和“陷入-模拟”的方式，即将Guest OS运行在非特权级，而将VMM运行于最高特权级（完全控制系统资源）。解除了Guest OS的特权级后，Guest OS的大部分指令仍可以在硬件上直接运行，只有执行到特权指令时，才会陷入到VMM模拟执行（陷入-模拟）。“陷入-模拟”的本质是保证可能影响VMM正确运行的指令由VMM模拟执行，大部分的非敏感指令还是照常运行。

因为x86指令中有若干条指令是需要被VMM捕获的敏感指令，但是却不是特权指令（称为临界指令），因此“特权解除”并不能导致它们发生陷入模拟，从而阻碍指令的虚拟化。x86下的敏感指令大致分类如下。

（1）访问或修改机器状态或虚拟机状态的指令。

(2) 访问或修改敏感寄存器或存储单元的指令，比如访问时钟寄存器和中断寄存器。

(3) 访问存储保护系统或内存、地址分配系统的指令（段页之类）。

(4) 所有I/O指令。

其中的(1)和(4)都是特权指令，在内核态下执行时会自动产生陷阱被VMM捕获，但是(2)和(3)不是特权指令，而是临界指令。部分临界指令会因为Guest OS的权限解除执行失败，但是却不会抛出异常，所以不能被捕获，如(3)中的VERW指令。

基于x86计算架构的指令处理原理，CPU的精细化调度技术采用了vCPU调度分配机制来实现精细化管控，通常也称之为CPU QoS功能（见图4-2）。

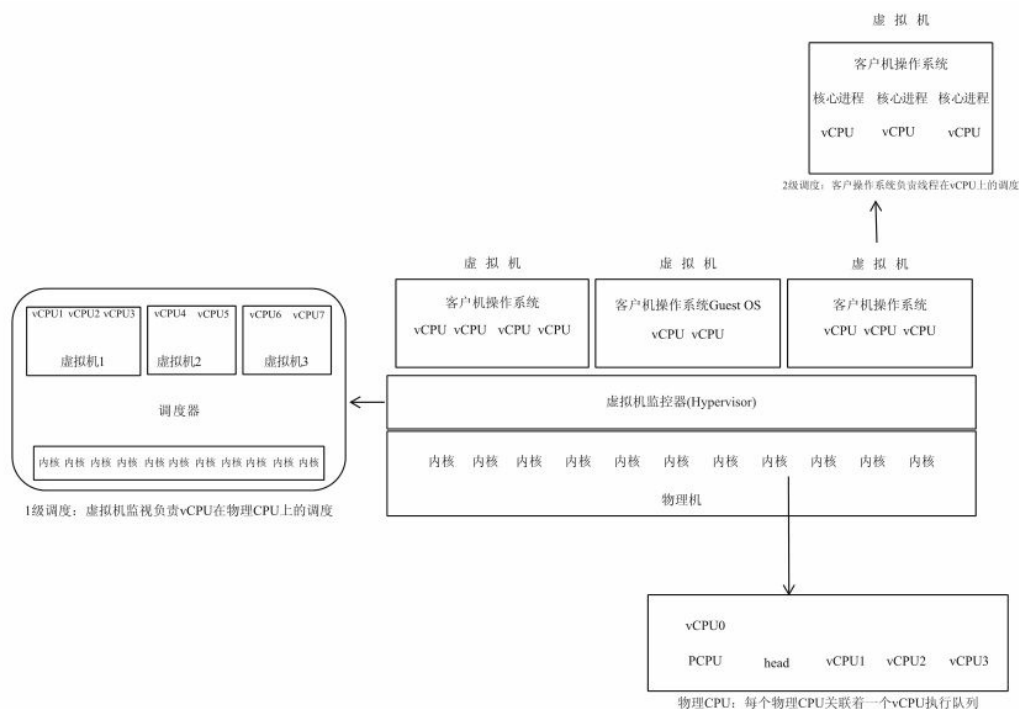


图4-2 vCPU调度分配机制

从虚拟机系统的结构与功能划分可以看出，客户操作系统与虚拟机监视器共同构成了虚拟机系统的两级调度框架，图4-2是一个多核环境下虚拟机系统的两级调度框架。客户操作系统负责第2级调度，即线程或进程在vCPU上的调度（将核心线程映射到相应的虚拟CPU上）。虚拟机

监视器负责第1级调度，即vCPU在物理处理单元上的调度。两级调度的调度策略和机制不存在依赖关系。vCPU调度器负责物理处理器资源在各个虚拟机之间的分配与调度，本质上把各个虚拟机中的vCPU按照一定的策略和机制调度在物理处理单元上，可以采用任意的策略来分配物理资源，满足虚拟机的不同需求。vCPU可以调度在一个或多个物理处理单元执行（分时复用或空间复用物理处理单元），也可以与物理处理单元建立一对一固定的映射关系（限制访问指定的物理处理单元）。

简言之，就是使得虚拟机获得的资源可以衡量，并且资源在虚拟机之间不相互干扰；这里的资源是计算资源、内存资源、网络资源和存储资源。

所以，CPU QoS功能实现的最终形态包括以下几种。

- 资源上限限制：适用资源严格隔离场景。
- 资源下限预留：适用资源衡量场景，结合上限设置实现资源可销售。
- 资源份额分配：适用资源复用场景。
- 上限限制：保证虚拟机隔离资源竞争，保证用户体验。
- 下限预留：最低资源保障，保证服务质量。
- 份额分配：针对用户划分不同等级，实现资源竞争。

CPU QoS的价值在于为应用提供计算服务质量的保障，确保针对资源的分配是确定的、可衡量的。

CPU QoS的三个特性带来的价值举例说明如下（假设资源为10份，有3台虚拟机）。

- 限制：VM1/VM2/VM3各设置资源限制为3份，则3台虚拟机最多只能使用各自3份资源，不会争抢其他虚拟机的份额资源或剩余的1份资源，实现严格资源隔离，保障用户体验。

- 预留：VM1/VM2/VM3各设置资源预留为3份，则3台虚拟机最少可以使用3份资源，保证资源销售的可衡量性，保障服务质量。
- 份额：VM1/VM2/VM3各设置资源份额为高/中/低，则3台虚拟机在充分竞争资源时，根据资源不同按照比例分配（见图4-3）。



图4-3 CPU QoS

CPU预留是排他性的，是资源设置预留的排他，而不是资源使用的排他：

- 以单台物理服务器节点上的CPU作为计算资源池，命名为pool；
- 上限限制不能超过这个pool的能力；
- 预留设置不能超过上限限制，如果上限没有设置，那么不能超过pool的能力；
- 预留对资源设置的排他性：虚拟化平台看到的底层资源是pool，如果给虚拟机VM1设置了预留pool1，那么要给VM2设置的预留资源量只能有pool-pool1这么多；

➤ 对预留的资源使用是共享复用的而不是排他的，就是说总资源是pool，VM1预留了pool1，VM2预留了pool2，其中 $pool1+pool2=pool$ ；如果VM1并没有使用完pool1的资源，那么VM2可以使用大于pool2的资源。

2. NUMA架构感知的调度技术

随着x86架构体系的发展，大部分用于提供计算资源的物理器件，也就是x86服务器，都按照SMP的系统架构来进行处理能力的增强。这就引入了NUMA的问题。

通过虚拟化软件的Host NUMA技术，可以显著提高虚拟机的性能，降低处理时延，以下针对NUMA和虚拟化软件的Host NUMA技术进行详细描述。

NUMA是非一致性内存架构（Non-uniform Memory Architecture），解决了SMP系统中的可扩展性问题。NUMA将几个CPU通过内存总线与一块内存相连构成一个组，整个系统就被分为若干个Node（见图4-4）。

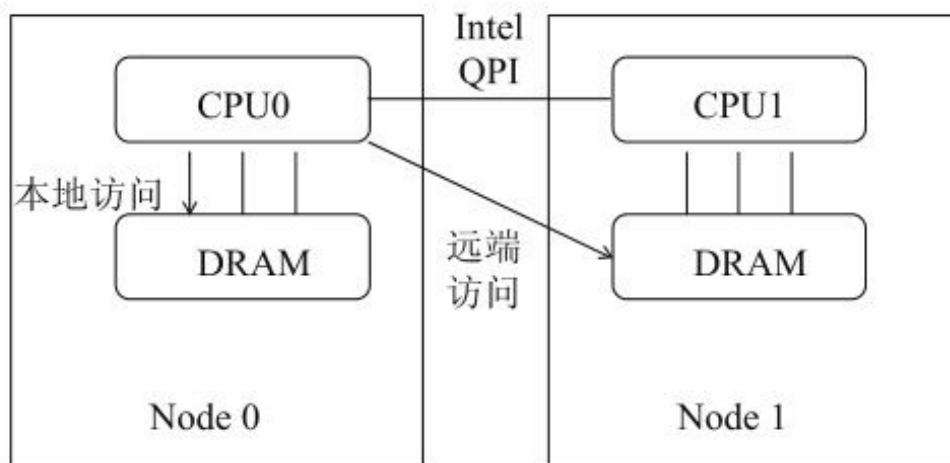


图4-4 CPU Node

一个Node服务器内包含若干CPU和一块内存，CPU访问其所在Node的本地内存的速度最快，访问其他Node的内存性能较差。

操作系统根据SRAT和SLIT表识别NUMA拓扑（见图4-5）。

虚拟化软件实现的Host NUMA主要提供CPU负载均衡机制，解决CPU资

源分配不平衡引起的VM性能瓶颈问题，当启动VM时，Host NUMA根据当时主机内存和CPU负载，选择一个负载较轻的Node放置该VM，使VM的CPU和内存资源分配在同一个Node上。如图4-5左边所示，Host NUMA把VM的物理内存放置在一个Node上，对VM的vCPU调度范围限制在同一个Node的物理CPU上，并将VM的vCPU亲和性绑定在该Node的物理CPU上。考虑到VM的CPU负载是动态变化的，在初始放置的Node上，Node的CPU资源负载也会随之变化，这会导致某个Node的CPU资源不足，而另一个Node的CPU资源充足，在此情况下，Host NUMA会从CPU资源不足的Node上选择VM，把VM的CPU资源分配在CPU资源充足的Node上，从而动态实现Node间的CPU负载均衡。

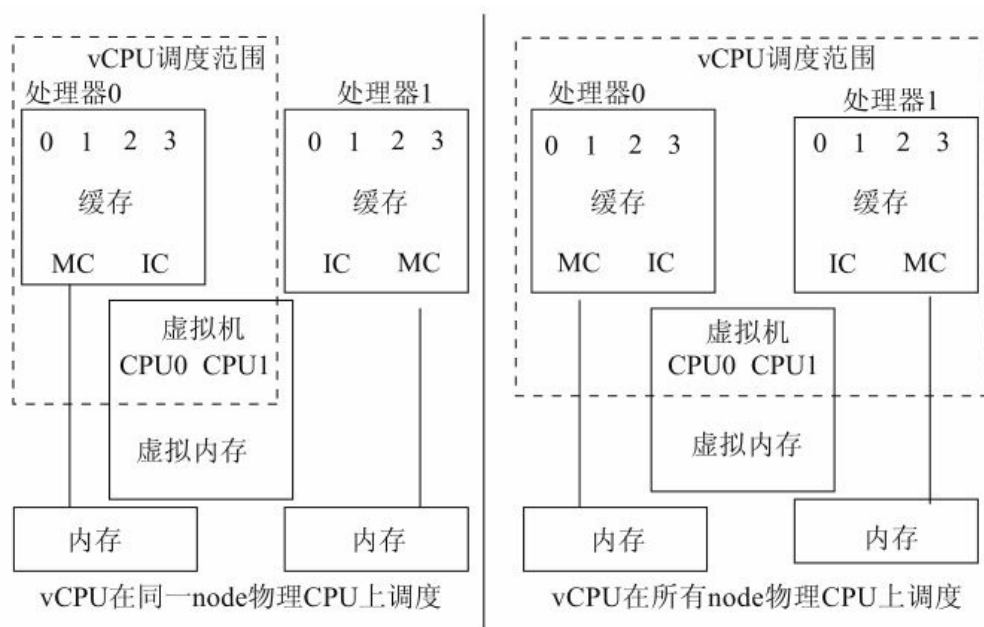


图4-5 Host NUMA原理图

对于VM的vCPU个数超过Node中CPU的核数的VM，如图4-5右边所示，Host NUMA把该VM的内存均匀地放置在每个Node上，vCPU的调度范围为所有Node的CPU。用户绑定了VM的vCPU亲和性，Host NUMA特性根据用户的vCPU亲和性设置决定VM的放置，若绑定在一个Node的CPU上，Host NUMA把VM的内存和CPU放置在一个Node上，若绑定在多个Node的CPU上，Host NUMA把VM的内存均匀分布在多个Node上，VM的vCPU在多个Node的CPU上均衡调度。

虚拟化软件提供复杂的NUMA调度程序来动态平衡处理器负载，根据当时主机内存和CPU负载优先把VM的CPU和内存资源分配在同一个Node

上，并随着资源负载的动态变化对主机Node间的CPU资源做负载均衡。

此特性为虚拟化软件基本特性，不需要管理员明确处理节点之间的虚拟机平衡，在各种场景均可使用。

Host NUMA保证VM访问本地物理内存，减少了内存访问延迟，可以提升VM性能，性能提升的幅度与VM虚拟机访问的内存大小和频率相关。

3. 内存复用技术

前文介绍了计算虚拟化技术中CPU分配的技术，本节针对提高资源利用率的内存管理和复用技术进行说明（见图4-6）。

计算虚拟化软件中，VMM（Virtual Machine Monitor）掌控所有系统资源，因此VMM掌握整个内存资源，负责页式内存管理，维护虚拟地址到机器地址的映射关系。因Guest OS本身亦有页式内存管理机制，则有VMM的整个系统就比正常系统多了一层映射。

映射关系如下：Guest OS: $PA = f(VA)$ 、VMM: $MA = g(PA)$ 。

VMM维护一套页表，负责PA到MA的映射。Guest OS维护一套页表，负责VA到PA的映射。实际运行时，用户程序访问VA1，经Guest OS的页表转换得到PA1，再由VMM介入，使用VMM的页表将PA1转换为MA1。

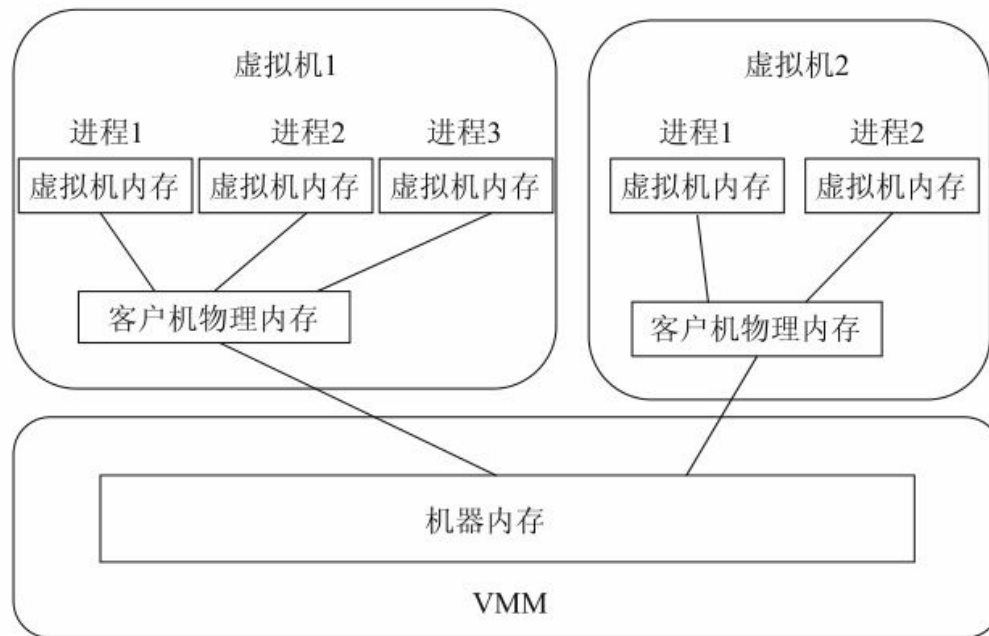


图4-6 内存虚拟化三层模型

页表虚拟化技术原理

普通MMU只能完成一次虚拟地址到物理地址的映射，在虚拟机环境下，经过MMU转换所得到的“物理地址”并不是真正的机器地址。若需得到真正的机器地址，必须由VMM介入，再经过一次映射才能得到总线上使用的机器地址。如果虚拟机的每个内存访问都需要VMM介入，并由软件模拟地址转换，效率是很低下的，几乎不具有实际可用性，为实现虚拟地址到机器地址的高效转换，现普遍采用的思想是：由VMM根据映射f和g生成复合的映射fg，并直接将这个映射关系写入MMU。当前采用的页表虚拟化方法主要是MMU类虚拟化（MMU Paravirtualization）和影子页表，后者已被内存的芯片辅助虚拟化技术所替代。

MMU Paravirtualization的基本原理是当Guest OS创建一个新的页表时，会从它所维护的空闲内存中分配一个页面，并向Xen注册该页面，Xen会剥夺Guest OS对该页表的写权限，之后Guest OS对该页表的写操作会陷入到Xen加以验证和转换。Xen会检查页表中的每一项，确保他们只映射了属于该虚拟机的机器页面，而且不得包含对页表页面的可写映射。然后Xen会根据自己所维护的映射关系，将页表项中的物理地址替换为相应的机器地址，最后再把修改过的页表载入MMU。如此，MMU就可以根据修改过的页表直接完成虚拟地址到机器地址的转换。

内存芯片辅助虚拟化

内存的芯片辅助虚拟化技术是用于替代虚拟化技术中软件实现的“影子页表”的一种芯片辅助虚拟化技术，其基本原理是：GVA（客户操作系统的虚拟地址）转换到GPA（客户操作系统的物理地址），然后再转换到HPA（宿主操作系统的物理地址）。两次地址转换都由CPU硬件自动完成（软件实现内存开销大、性能差）。我们以VT-x技术的页表扩充技术Extended Page Table（EPT）为例。首先，VMM预先把客户机物理地址转换到机器地址的EPT页表设置到CPU中；其次，客户机修改客户机页表无需VMM干预；最后，地址转换时，CPU自动查找两张页表完成客户机虚拟地址到机器地址的转换。使用内存的芯片辅助虚拟化技术，客户机运行过程中无需VMM干预，去除了大量软件开销，内存访问性能接近物理机。

虚拟化软件平台提供多种内存复用技术和灵活自动的内存复用策略。对于某些物理内存资源比较紧张的场景，如果用户希望运行超过物理内存能力的虚拟机，以达到节省成本的目的，就需要有内存复用策略来动态地对内存资源进行分配和复用。内存复用策略通过内存复用技术，提升物理内存利用率的同时，尽可能减少对虚拟机性能的影响。客户无需关心何时调用和怎么调用几种复用技术，只需简单配置和开启复用策略后就能达到提升虚拟机密度的目的。

虚拟化软件的内存复用技术有以下三种：内存气泡、内存零页共享和内存交换技术。

（1）内存气泡技术（Ballooning）

内存气泡技术是一种VMM通过“诱导”客户机操作系统来回收或分配客户机所拥有的宿主机物理内存的技术。当客户机物理内存足够时，客户机操作系统从其闲置客户机物理机内存链表中返回客户机物理内存给气球；当客户机物理内存资源稀缺时，客户机操作系统必须回收一部分客户机物理内存，以满足气球申请客户机物理内存的需要。通过Balloon Driver模块，从源虚拟机申请可用内存页面，通过Grant Table授权给目标虚拟机，并更新虚拟机物理地址和机器地址映射关系表。

通过使用Ballooning技术，可以提升内存使用效率。

（2）内存零页共享技术

内存零页共享技术作为内存复用技术的一种，能有效地识别和释放虚拟机内未分配使用的零页，以达到提高内存复用率的目的。客户开启零页共享技术后，能实时从虚拟机内部把零页进行共享，从而把其占用的内存资源释放出来给其他虚拟机使用，以创建更多的虚拟机，实现提高虚拟机密度的目的。与内存气泡技术不同，零页共享后的内存页对于虚拟机来说还是可用的，虚拟机可以随时根据需要收回这部分内存，用户体验相对来说更加友好。

用户进程定时扫描虚拟机的内存数据，如果发现其数据内容全为零，则通过修改P2M映射的形式把其指向一个特定的零页，从而做到在物理内存中仅保留一份零页拷贝。虚拟机的所有零页均指向该页，以达到节省内存资源的目的。当零页数据发生变动时，由Xen动态地分配一页内存出来给虚拟机，使修改后的数据有内存页进行存放，因此对于Guest OS来说，整个零页共享过程是完全不感知的。

（3）内存交换技术

内存交换技术作为内存复用技术的一种，能通过Xen把虚拟机内存数据交换出到存储介质上的交换文件中，从而释放内存资源，以达到提高内存复用率的目的。由于内存气泡和零页共享的数量与虚拟机本身的内存使用情况强相关，因此其效果不是很稳定，用户使用内存交换技术，可以弥补上述不足，可以保证释放一定量的内存空间（理论上所有虚拟机内存都能交换出来），但同时会带来一定程度的虚拟机性能下降。

内存交换触发时，根据用户需要告知Xen需要向某个虚拟机交换出一定量的内存页出来，Xen按一定的选页策略从虚拟机中选择相应数量的页后，把页数据保存到存储介质上的交换文件中，同时释放原先存放数据的那些页供其他虚拟机使用。当虚拟机读写的页正好是被换出的页时，在缺页处理时Xen会重新为其分配一页内存，然后从存储介质上的交换文件中把相应的页交换回新分配的内存页中，同时再选择另外一页内存交换出去，从而保证虚拟机对页的正常读写的同时，稳定交换页的数量。这个过程与零页共享一样，对Guest OS都是不可感知的。

4. I/O调度中断优化技术

影响电信和企业关键应用的运行时响应时延的因素很多，其中，中断处

理就是一个在虚拟化条件下的关键因素。

CPU在处理I/O访问请求出现中断时，其实际处于等待的状态，在虚拟化的条件下，因为单个物理机上运行的虚拟机实例比较多，因此所产生的I/O中断请求也变多，这样实际上造成了物理CPU的等待，从而影响了应用运行的响应时延和性能。

VMM通过I/O虚拟化来复用有限的外设资源，其通过截获Guest OS对I/O设备的访问请求和通过软件模拟真实的硬件来响应这些截获的请求，目前I/O设备的虚拟化方式主要有三种：设备接口完全模拟、前端/后端模拟、直接划分。

设备接口完全模拟

设备接口完全模拟即软件精确模拟与物理设备完全一样的接口，Guest OS驱动无需修改就能驱动这个虚拟设备。优点是没有额外的硬件开销，可重用现有驱动程序，缺点是为完成一次操作要涉及多个寄存器的操作，使得VMM要截获每个寄存器访问并进行相应的模拟，导致多次上下文切换，性能较低。

前端/后端模拟

VMM提供一个简化的驱动程序（后端，Back-End），Guest OS中的驱动程序为前端（Front-End，FE），前端驱动将来自其他模块的请求通过与Guest OS间的特殊通信机制直接发送给Guest OS的后端驱动，后端驱动在处理完请求后再发回通知给前端（Xen即采用该方法）。优点是出于基于事务的通信机制，能在很大程度上减少上下文切换开销，没有额外的硬件开销，缺点是需要VMM实现前端驱动，后端驱动可能成为瓶颈。

直接划分

直接划分即直接将物理设备分配给某个Guest OS，由Guest OS直接访问I/O设备（不经VMM），目前与此相关的技术有AMD IOMMU、Intel VT-d、PCI-SIG之SR-IOV等，旨在建立高效的I/O虚拟化直通道。优点是直接访问减少了虚拟化开销，缺点是需要购买额外的硬件。

5. 网络直通VMDq技术

在虚拟化的条件下，网络访问报文需要从软件层的虚拟网卡经过物理网卡才能发出。由于软件模拟虚拟网卡的因素会造成网络访问时延的增加和抖动，为了解决这个问题，我们先后发展出VMDq的技术和SR-IOV的技术。

在虚拟环境中，Hypervisor管理网络I/O活动，随着平台中的虚拟机和传输量增加，Hypervisor需要更多的CPU周期来进行数据包分类操作，并需要将数据向路由到相应的虚拟机中，这些操作会因对CPU的占用，而影响上层应用软件对CPU的使用。Hypervisor利用VMDq（Virtual Machine Device Queues，虚拟机设备队列）技术，针对对虚拟机网络性能有极高要求的场景，在支持VMDq的网卡上，用硬件实现Layer 2分类/排序器，根据MAC地址和VLAN信息将数据包发送到指定的网卡队列中。这样虚拟机收发包时就不需要DomO的参与，这种模式极大地提升了虚拟化网络效率。

Intel VMDq技术，是专门用于提升网卡的虚拟化I/O性能的硬件辅助I/O虚拟化技术，主要解决I/O设备上频繁的VMM切换以及对中断的处理问题，其可以减轻Hypervisor的负担，同时提高虚拟化平台网络I/O性能。

VMDq技术可以将网络I/O管理负担从Hypervisor上卸载掉，多个队列和芯片中的分类智能性支持虚拟环境中增强的网络传输流，从应用任务中释放处理器周期，提高向虚拟机的数据处理效率及整体系统性能。

VMDq为虚拟机提供接近物理机的网络通信性能，兼容部分虚拟化高级特性，比如在线迁移、虚拟机快照等。

6. 网络直通SR-IOV技术

与VMDq类似，SR-IOV也是采用类直通的方式来避免软件层对于网络转发的时延、抖动的影响，从而满足电信与企业关键应用对于高性能低时延的要求。

服务器虚拟机技术是通过软件模拟多个网络适配器的方式来共享一个物理网络适配器端口，来满足虚拟机的I/O需求。虚拟化软件在多个层面控制和影响虚拟机I/O操作，因此导致环境中出现瓶颈并影响I/O性能。SR-IOV是一种不需要软件模拟就可以共享I/O设备I/O端口的物理功能的方法，主要利用iNIC实现网桥卸载虚拟网卡，允许将物理网络适配器的SR-IOV虚拟功能直接分配给虚拟机，可以提高网络吞吐量，并缩短网络延迟，同时减少处理网络流量所需的主机CPU开销。

SR-IOV (Single Root I/O Virtualization) 是PCI-SIG推出的一项标准，是虚拟通道（在物理网卡上对上层软件系统虚拟出多个物理通道，每个通道具备独立的I/O功能）的一个技术实现，用于将一个PCIe设备虚拟成多个PCIe设备，每个虚拟PCIe设备如同物理PCIe设备一样向上层软件提供服务。通过SR-IOV，一个PCIe设备不仅可以导出多个PCI物理功能，还可以导出共享该I/O设备上的资源的一组虚拟功能，每个虚拟功能都可以被直接分配到一个虚拟机，能够让网络传输绕过软件模拟层，直接分配到虚拟机，实现将PCI功能分配到多个虚拟接口以在虚拟化环境中共享一个PCI设备的目的，并且降低了软件模拟层中的I/O开销，因此实现了接近本机的性能。在这个模型中，不需要任何透传，因为虚拟化在终端设备上发生，允许管理程序简单地将虚拟功能映射到VM上以实现本机设备性能和隔离安全。SR-IOV虚拟出的通道分为两个类型。

- PF (Physical Function) 是完整的PCIe设备，包含了全面的管理、配置功能，Hypervisor通过PF来管理和配置网卡的所有I/O资源。

- VF (Virtual Function) 是一个简化的PCIe设备，仅仅包含了I/O功能，通过PF衍生而来，好像物理网卡硬件资源的一个切片。对于Hypervisor来说，这个VF同一块普通的PCIe网卡一模一样，可满足高网络I/O应用要求，无需特别安装驱动，且可以实现无损热迁移、内存复用、虚拟机网络管控等虚拟化特性。

4.2 跨服务器的计算资源调度算法

在云计算采用计算虚拟化技术实现了虚拟机的模拟和虚拟机运行实例的隔离需求之后，更重要的是要把大量计算虚拟化资源组合成一个大资源池，用来满足云化场景下资源的按需申请和资源的灵活分配的要求。这就引入了新的课题，即如何将虚拟机在这个大资源池（包括跨地域场景）进行高效调度。

4.2.1 高性能、低时延的虚拟机热迁移机制

虚拟机是弹性计算服务的资源实体，为保证在虚拟资源池中对虚拟机资源的灵活分配，需提供在资源池内的虚拟机热迁移能力，即虚拟机在不中断业务的情况下实现在不同物理机上的迁移。虚拟机迁移时，管理系

统会在迁移的目的端创建该虚拟机的完整镜像，并在源端和目的端进行同步。同步的内容包括内存、寄存器状态、堆栈状态、虚拟CPU状态、存储以及所有虚拟硬件的动态信息。在迁移过程中，为保证内存的同步，虚拟机管理器（Hypervisor）提供了内存数据的快速复制技术，从而保证在不中断业务的情况下将虚拟机迁移到目标主机。同时，通过共享存储，保证了虚拟机迁移前后持久化数据不变。

虚拟机热迁移的作用，具体如下。

- 降低客户的业务运行成本：根据时间段的不同，客户的服务器会在一定时间内处于相对空闲的状态，此时若将多台物理机上的业务迁移到少量或者一台物理机上运行，而将没有运行业务的物理机关闭，就可以降低客户的业务运行成本，同时达到节能减排的作用。
- 保证客户系统的高可靠性：如果某台物理机运行状态出现异常，在进一步恶化之前将该物理机上运行的业务迁移到正常运行的物理机上，就可以为客户提供高可用性的系统。
- 硬件在线升级：当客户需要对物理机硬件进行升级时，可先将该物理机上的所有虚拟机迁移出去，之后对物理机进行升级，升级完成再将所有虚拟机迁移回来，从而实现在不中断业务运行的情况下对硬件进行升级，保证服务的持续可用性。

一个虚拟化系统，至少需要在下列场景下支持虚拟机热迁移功能：

- 根据需要按照迁移目的手动把虚拟机迁移到空闲的物理服务器；
- 根据资源利用情况将虚拟机批量迁移到空闲的物理服务器。

虚拟机应用于电信和企业关键应用领域，且虚拟网元在硬件平台间无缝迁移时，面临切换时延带来虚拟机内部业务中断的问题，通信领域虚拟机迁移切换时间确保在1秒以内，同时电信和企业关键应用业务压力大，对热迁移构成挑战。一般可以采用如下技术提升热迁移的效果。

混合拷贝热迁移，具体如下。

- Precopy和Postcopy按需自适应切换，避免重载业务无限制迭代拷贝。
- 前台和后台数据传输带宽管控，充分利用带宽完成后台传输。
- 识别热点内存，减少缺页概率。

RDMA/RoCE热迁移加速的具体方法如下：

- 通过RDMA硬件能力加速，提升迁移效率；
- 以vMotion与RDMA相结合的方式加速；
- 采用RoCE网卡直通技术；
- 在Host中集成RoCE协议栈，改造热迁移等服务采用的协议栈。

4.2.2 计算资源池的动态资源调度管理和动态能耗管理

在云化资源池的环境下，虚拟机的负载是动态变化的，因此计算资源池的管理系统需要实现对于虚拟机部署位置的自动化调整，以保障虚拟机能够获得所需要的资源，同时也保障资源池内的负载是均衡的，从而保障资源池资源的高效利用。

实现该技术的方式，通常称为动态资源调度管理，简称为DRS（Dynamic Resource Schedule）。

DRS周期性监控集群下的物理主机和虚拟机负载（CPU和内存），如果主机负载的不均衡度（标准方差）超过集群配置的阈值，则触发虚拟机迁移，使得集群范围内的负载区域平衡。作为DRS的扩展功能，DPM（Distributed Power Management，分布式电源管理）支持在集群负载低（CPU和内存负载小于集群配置的阈值）时，主动对某些主机进行下电，达到节能减排的目的，同时在集群负载高（负载大于集群配置的阈值）时，重新启动一些主机，满足业务需要。

以下是DRS的几种基本的应用场景。

场景1：集群内主机间的负载不均衡，调度后将部分虚拟机从负载高的主机迁移至负载低的主机，达到负载均衡。

场景2：集群内主机间的负载不均衡，负载也较低，调度后负载达到平衡，同时对经过评估可以下电的主机（物理主机3）进行下电。

场景3：集群内主机的负载均很高，调度后将处于已下电状态的主机（物理主机3）上电，并将部分虚拟机迁移至负载低的主机上，达到负载均衡。

实现DRS和DPM的调度算法很多，通常衡量一个资源池内部的负载不平衡的程度，可以采用每个计算服务器节点的负载与资源池平均负载差异值的标准方差来衡量。差值越大，表示该节点与资源池平均负载的差异越大，越需要调整其负载，以保证尽快将集群的负载进行均衡。

关于进行资源平衡调整的方法，业界有很多的实现方式。其会针对不同的负载波动水平，不同的业务可靠性要求，在集群负载收集的时间和门限、计算节点负载调整的门限和时间间隔等处理上进行不同的调整。比较优良的算法还会保证根据历史的运行状况进行负载预测，以避免负载调整的震荡。

通常情况下，动态资源调整主要考虑CPU和内存的负载变化情况，但网络 and 存储的负载在面对电信和企业关键应用中也是重要的且必须考虑的因素，针对CPU/内存/网络/存储等多维资源的负载预测可以采用以上类似算法。

真正进行负载调整时，进行业务部署调整的原则为：负载方差大于阈值，或物理机利用率超过阈值。

需要进行部署位置调整的源VM选择原则为：选择离均值最远的物理主机之上的“最有效”VM（即迁移后使该物理主机的负载最靠近均值）。

目标物理主机选择的原则：选择接受源VM后方差下降最多的物理主机。

通过重复步骤计算，直到方差低于阈值或者达到VM迁移次数限制。

需要说明的是，对于虚拟机和物理机的负载计算方法，动态能耗管理的

方法实际上与动态资源调度管理的方法是一致的，主要的区别在于动态能耗管理需要尽量地合并物理主机上的虚拟机业务，以尽量少的物理机占用达成节省能源消耗的目的。

4.3 计算高可靠性保障

在云计算的条件下，因业务运行所需要的资源是通过软件模拟或者软件管理分配的方式提供的，也就是说在业务运行负载和物理硬件之间引入了Hypervisor作为管理层，保证业务运行的可靠性。

在通过虚拟化技术保障云计算条件下的业务运行可靠性方面，当前业界发展出来的技术主要有以下两种：①基于冷备机制的虚拟化HA保护，这种方式主要提供了在物理硬件故障的条件下，选择资源池中的其他健康物理主机重新部署虚拟机，并在原有数据不丢失的条件下，尽快恢复虚拟机业务；②基于虚拟机热备机制的虚拟机运行态镜像冗余方案，这种方式主要是在不同的物理主机上，提供虚拟机业务运行的镜像冗余，在一台物理主机发生故障的时候，自动由另外一台主机上的虚拟机业务运行镜像接管进行业务处理，从而保证虚拟机的业务不发生中断。

上述两种方式实现的技术难度和工程化部署的要求限制不一样。其中冷备机制实现技术难度较低，对工程化部署的限制和要求较少，在业界得到广泛的应用。热备机制的实现难度较大，同时，因需要提供虚拟机运行业务镜像冗余，虚拟机的性能会有额外的占用，同时也需要在工程化部署时保证虚拟机的业务组网能够在切换的条件下平滑过渡，对于业务组网的要求较高，所以现阶段的应用还处于逐步走向成熟的过程。

4.3.1 基于冷备机制的虚拟机HA保护

当物理服务器宕机或者重启时，系统可以将具有HA属性的故障虚拟机迁移到其他物理服务器，保证虚拟机快速恢复。

由于单个集群内可以运行上千个虚拟机，当某个或某些服务器宕机后，为避免大量虚拟机迁移造成网络拥塞和目的服务器过载，系统会根据网络流量、目的服务器负荷选择将虚拟机迁移到不同的目的服务器。

当VRM（Virtual Resource Manager，虚拟资源管理器）与物理服务器上的计算代理心跳中断超过30秒时，会触发虚拟机HA，当一个虚拟机由

运行状态突然异常消失时，也会触发HA在其他正常的计算节点上快速恢复业务。

通过存储层面的锁机制可防止同一个虚拟机实例在多个物理机器上同时启动。

当一个物理服务器节点掉电恢复后，业务进程开机自启动恢复，之前运行的虚拟机全部故障迁移至其他物理节点。

4.3.2 基于热备机制的虚拟机运行业务镜像冗余方案

在虚拟环境下设置主备虚拟机，在备节点上创建主虚拟机的完整拷贝。主节点上虚拟机的CPU状态、内存、磁盘操作、QEMU等与备节点虚拟机保持低延迟的定时同步。备节点虚拟机定时检测主节点虚拟机心跳，在指定时间内收不到心跳即认为异常发生，备虚拟机切换到正常运行状态。这个方案的优势是主备节点可以保持状态完全同步，数据完全一致，缺点是会带来一些性能开销。

4.3.3 无状态计算及物理机可靠性保障

虚拟机的本质就是通过虚拟化技术，将一台物理服务器虚拟成多个计算机。虚拟机之间彼此相互独立，一个虚拟机故障不会影响其他虚拟机。用户对虚拟机的使用体验和对传统物理机的体验相同。

在一个虚拟机内的任何操作，不会对同一台物理服务器上的其他虚拟机和虚拟化平台自身的可用性产生危害。即使虚拟机的运行出现故障，比如操作系统崩溃、应用程序错误导致死机等情况，同一物理服务器上的虚拟化平台以及其他虚拟机仍然可以正常运行，继续为用户提供服务。

第5章 面向网络自动化、多租户的网络虚拟化

5.1 网络虚拟化的驱动力与关键需求

网络虚拟化的驱动力

服务器和存储虚拟化技术的迅猛发展，使得动态快速分配计算资源和存储资源成为很平常的事，从而大大缩短了创建虚拟服务器的时间。相比之下，目前的传统网络架构明显落后于虚拟化的要求，成为整个资源分配流程中的短板，表现在如下方面。

- 传统网络在虚拟化场景下应用部署效率低：在很多情况下提供网络资源需要手工配置、网络交换机端口配置、ACL、路由，等等。网络和应用安全策略的部署仍然需要手工配置，自动化程度很低。
- 网络变更困难，每次人工配置都需要小心翼翼，需要网络专家花费大量的时间来连接不通的设备，任何小的疏忽都可能造成网络故障。应用的部署也不得不拖延几天、几周甚至更长时间，直到网络资源最终准备好了为止。
- 网络没有移动性：网络的配置绑定于硬件。网络配置的状态遍及大量独立的网络设备（物理的和虚拟的）。底层VLAN、网关、防火墙等物理资源的部署限制了计算资源的部署与自由迁移。
- 不能充分利用网络资源：大多数公司能利用到网络资源的30%~40%，很多情况下大量资源闲置，而某些时候由于数据量周期性的猛增，又造成网络资源不够。很多ISP或者通讯网络提供商们很头痛，在用户基本收费增长不大的情况下，用户期望的数据流量却比以往增大很多。除了网络扩容之外，更重要的是提高目前已有网络的利用率（见图5-1）。

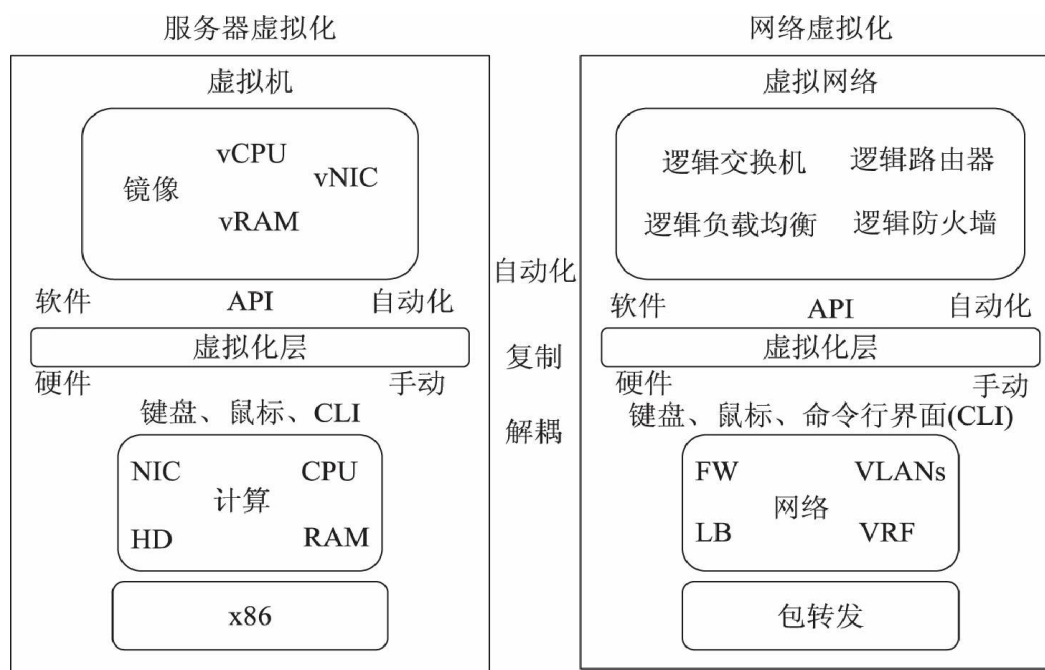


图5-1 网络虚拟化与服务器（计算）虚拟化类比

对网络虚拟化的关键需求

(1) 与物理层解耦：网络虚拟化的目标是接管所有的网络服务、特性和应用的虚拟网络必要的配置（VLANs、VRFs、防火墙规则、负载均衡池&VIPs、IPAM、路由、隔离、多租户等），从复杂的物理网络中抽取出简化的逻辑网络设备和服务，将这些逻辑对象映射给分布式虚拟化层，通过网络控制器和云管理平台的接口来消费这些虚拟网络服务，从而使应用只需和虚拟化网络层打交道，将复杂的网络硬件变为“隐士”，不给IT增加烦恼。

(2) 网络服务抽象：虚拟网络层可以提供逻辑端口、逻辑交换机和路由器、分布式虚拟防火墙、虚拟负载均衡器等，并可同时确保这些网络设备和服务的监控、QoS和安全。这些逻辑网络对象就像服务器虚拟化出来的vCPU和内存一样，可以和任意安全策略自由组合成任意拓扑的虚拟网络。

(3) 网络按需自动化：通过API自动化部署，一个完整的、功能丰富的虚拟网络可以自由定义任何约束在物理交换基础上的设施功能、拓扑或资源。通过网络虚拟化，每个应用的虚拟网络和安全拓扑就拥有了移动性，同时实现了和流动的计算层绑定，并且可通过API自动部署，又确

保了和专有物理硬件解耦。

（4）支持多租户网络安全隔离：计算虚拟化使多种业务或不同租户资源共享同一个数据中心资源，但其同时需要为多租户提供安全隔离网络。

5.2 SDN架构

解决云数据中心网络问题存在不同的方式，SDN（Software-Defined Networking）是其中之一。SDN并不是具体的技术，而是一种全新的网络设计框架，SDN的核心理念是改变传统网络对数据流的控制方式。在传统网络中，网络采用分布式控制面，报文从源到目的的转发行为由各个网络节点自己独立控制和完成，每个网络节点都需要独立的配置。SDN框架中的网络，控制面与转发面是分离的，转发面与具体协议无关（见图5-2、图5-3）。

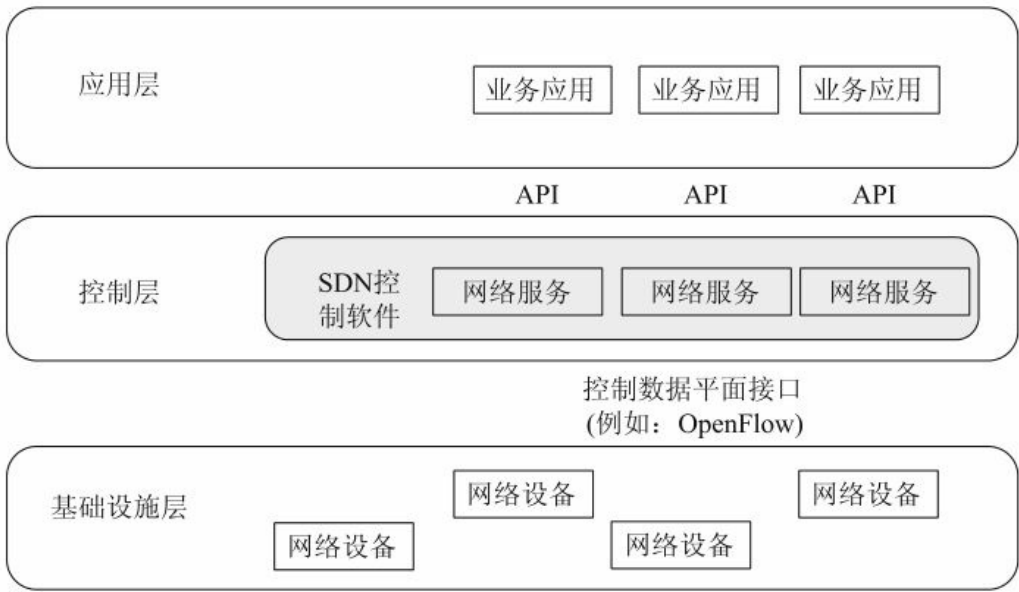


图5-2 SDN架构

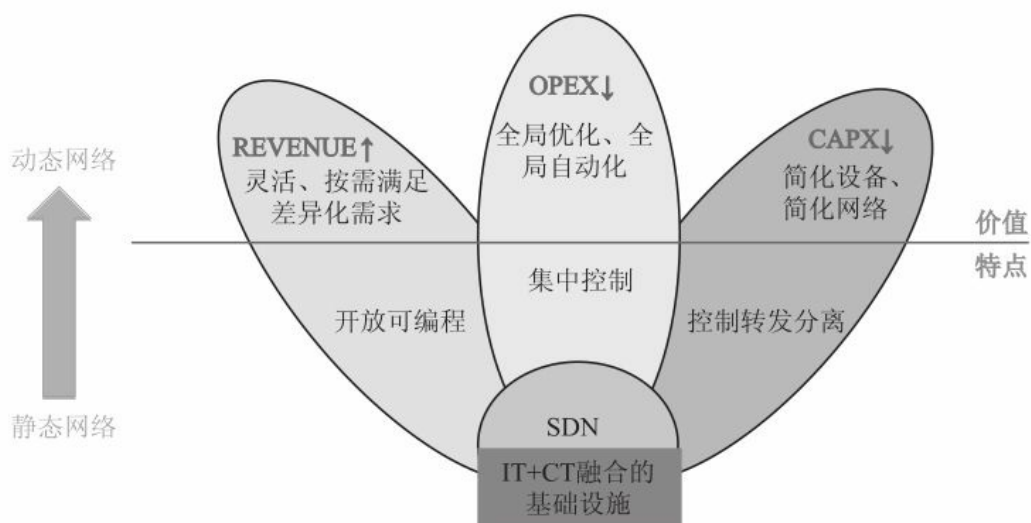


图5-3 SDN特点与价值

- (1) 控制面与转发面分离：控制面更灵活，转发面抽象与标准化。
- (2) 集中化的网络控制：控制软件具备全局网络视图，策略转发规则集中。
- (3) 网络开放可编程。
- (4) 网络业务的自动化应用程序控制。

对于SDN业界并无标准的理解，具体含义与其运行的网络领域和其使用的策略和协议相关。以下是几种SDN思路，如图5-4所示。

Open Networking Foundation	OpenFlow 控制转发分离	网络革新路线 技术难度高，各场景适配复杂
IETF	OpenAPI 控制面开放	网络优化路线 网管优化、提高自动化能力，但 开放不足；客户开发难度大
云计算软件厂商 (如VMWARE)	Overlay/NVo3 虚拟逻辑网络	IT解耦路线 IT新思路，当前主要针对DC内虚拟 化场景，缺乏统一网络资源管理
ETSI	NFV，网络功能虚拟化	网络设备软化路线 运营商新思路局：部优化网关功能 通用性，关键在标准与部署

图5-4 SDN路线

5.2.1 IETF定义的SDN架构介绍

IETF作为传统网络架构的制定者，其核心思路是重用当前的技术而不是OpenFlow，并关注重点设备控制面的功能与开放API。

XML-based SDN（Software-Defined Networking），使用Netconf等设备存在的接口对现有设备进行配置，对当前设备不修改（见图5-5）。

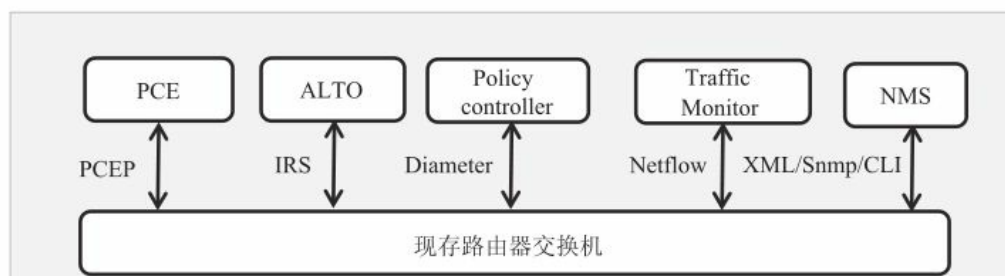


图5-5 I2RS与其他现存设备接口

IETF I2RS（Interface to Routing System：路由系统接口）工作组希望将路由协议中的策略配置运行在集中的控制器上，控制器通过设备反馈的事件、路径拓扑和网络流量信息来动态下发路由状态、策略到设备上，

具体的路由计算还是由各个网络设备分布式完成。

5.2.2 ONF OpenFlow网络架构

OpenFlow起源于斯坦福大学的Clean Slate项目组。该项目的最终目的是重新发明互联网，旨在改变设计已略显不合时宜，且难以进化发展的现有网络基础架构。在2006年，斯坦福的学生Martin Casado领导了一个关于网络安全与管理的项目Ethane，该项目试图通过一个集中式的控制器，让网络管理员可以方便地定义基于网络流的安全控制策略，并将这些安全策略应用到各种网络设备中，从而实现对整个网络通信的安全控制。受此项目（及Ethane的前续项目Sane）启发，Martin和他的导师Nick McKeown教授（时任Clean Slate项目的Faculty Director）发现，如果将Ethane的设计更一般化，将传统网络设备的数据转发（Data Plane）和路由控制（Control Plane）两个功能模块相分离，通过集中式的控制器（Controller）以标准化的接口对各种网络设备进行管理和配置，那么这将为网络资源的设计、管理和使用提供更多的可能性，从而更容易推动网络的革新与发展。于是，他们便提出了OpenFlow的概念，并且Nick McKeown等人于2008年在ACM SIGCOMM（美国计算机协会数据通信专业组会议）发表了题为“OpenFlow: Enabling Innovation in Campus Networks”的论文，首次详细地介绍了OpenFlow的概念。

SDN的设计理念是将网络的控制面与数据转发面进行分离，并实现可编程化控制。SDN的典型架构共分三层，最上层为应用层，包括各种不同的业务和应用；中间的控制层主要负责处理数据平面资源的编排，维护网络拓扑、状态信息等，在OpenFlow环境中，控制器会使用OpenFlow协议和Netconf协议与交换机联系（OpenFlow是将流数据发送到交换机的API，而NETCONF是网络配置API）。最底层的基础设施层负责进行基于流表的数据处理、转发和状态收集（见图5-6）。

SDN本质上具有“控制和转发分离”、“设备资源虚拟化”和“通用硬件及软件可编程”三大特性，这带来了一系列的好处。

第一，设备硬件归一化，硬件只关注转发和存储能力，与业务特性解耦，可以采用相对廉价的商用架构来实现。

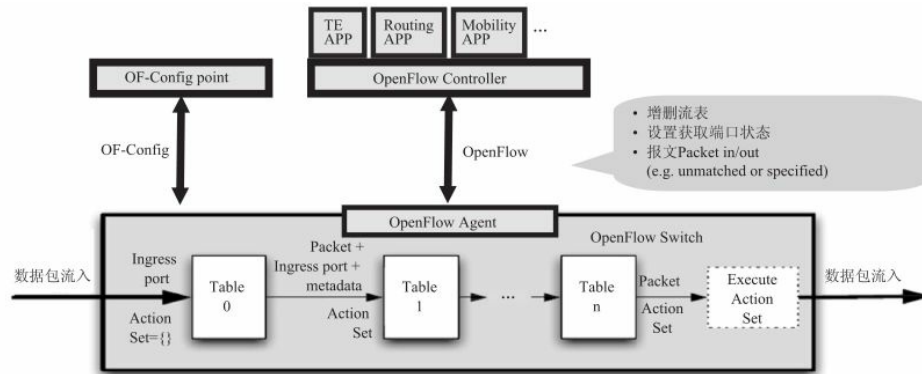


图5-6 基于OpenFlow的控制与转发分离体系结构

第二，网络的智能性全部由软件实现，网络设备的种类及功能由软件配置而定，对网络的操作控制和运行由服务器作为网络操作系统（NOS）来完成。

第三，对业务响应相对更快，可以定制各种网络参数，如路由、安全、策略、QoS、流量工程等，并实时配置到网络中，开通具体业务的时间将缩短。

ONF SDN的三大要素，具体如下。

（1）转发与控制分离，这使得网络交换机的数据转发变得更加简单、快速；同时，控制变成了网络操作系统中一个相对集中的逻辑功能。

（2） OpenFlow协议，它向交换机传送转发表，交换机依此转发报文。这种做法与传统网络完全不同。在传统网络架构中，交换机和路由器需要自己决定报文的转发路径，这可能会给网络运营商带来一些不可预知的负面影响，包括成本增加、性能降低、上市时间延缓等。有了SDN，控制软件决定报文的转发路径，使得运营商可以“随心所欲”地控制网络。

（3）具有一致性的、全系统范围的网络操作系统可编程接口，它能让网络实现真正意义上的可编程或者软件定义。如果不能实现转发与控制分离，那么几乎所有SDN所能带来的好处都无法体现；如果能实现转发和控制分离，但没有OpenFlow协议，那么就需要通过其他途径，将所需要的流量表信息传递给交换机。OpenFlow就是实现这一功能的行业标准。

OpenFlow/SDN吸引了业界越来越多的关注，成为近年来名副其实热门技术。目前，包括HP、IBM、Cisco、NEC以及国内的华为和中兴等在内的传统网络设备制造商都已纷纷加入OpenFlow的阵营，同时有一些支持OpenFlow的网络硬件设备已经面世。2011年，开放网络基金会（Open Networking Foundation）在Nick等人的推动下成立，专门负责OpenFlow标准和规范的维护和发展；同年，第一届开放网络峰会（OpenNetworking Summit）召开，为OpenFlow和SDN在学术界和工业界都做了很好的介绍和推广。2013年召开的第二届峰会上，来自Google的Urs Hölzle在以“OpenFlow@Google”为题的Keynote演讲中宣布Google已经在其全球各地的数据中心骨干网络中大规模地使用OpenFlow/SDN，从而证明了OpenFlow不再是仅仅停留在学术界的一个研究模型，而是已经完全具备了可以在产品环境中应用的技术成熟度。最近，Facebook也宣布其数据中心中使用了OpenFlow/SDN的技术。

5.2.3 OpenFlow协议介绍

自2010年初发布第一个版本（v1.0）以来，OpenFlow规范已经经历了1.1、1.2、1.3、1.4等版本。同时，OpenFlow管理和配置协议也发布了第一个版本（OF-Config 1.0 & 1.1）。

OF规范主要分为如下四大部分。

（1）OpenFlow的端口（Port）

OpenFlow规范将Switch上的端口分为三类别：

- 物理端口，即设备上物理可见的端口；
- 逻辑端口，在物理端口基础上由Switch设备抽象出来的逻辑端口，如为tunnel或者聚合等功能而实现的逻辑端口；
- OpenFlow目前总共定义了ALL、CONTROLLER、TABLE、IN_PORT、ANY、LOCAL、NORMAL和FLOOD这8种端口，其中后三种为非必需的端口，只在混合型的OpenFlow Switch（OpenFlow-hybrid Switch，即同时支持传统网络协议栈和OpenFlow协议的Switch设备，相对于OpenFlow-only Switch而言）中存在。

（2）OpenFlow的FlowTable（国内直译为“流表”）

OpenFlow通过用户定义的或者预设的规则来匹配和处理网络包。一条OpenFlow的规则由匹配域（Match Fields）、优先级（Priority）、处理指令（Instructions）、统计数据（如Counters）、超时时间（Timeout）、附属属性（Cookie）等字段组成，如图5-7所示。

Mach Fields	Priority	Counters	Instructions	Timeouts	Cookie
-------------	----------	----------	--------------	----------	--------

图5-7 OpenFlow规则

在一条规则中，可以根据网络包在L2、L3或者L4等网络报文头的任意字段进行匹配，比如以太网帧的源MAC地址，IP包的协议类型和IP地址，或者TCP/UDP的端口号等。目前OpenFlow的规范中还规定了Switch设备厂商可以选择性地支持通配符进行匹配。据说，OpenFlow在未来还计划支持对整个数据包的任意字段进行匹配。

所有OpenFlow的规则都被组织在不同的FlowTable中，在同一个FlowTable中按规则的优先级进行先后匹配。一个OpenFlow的Switch可以包含一个或者多个FlowTable，从0依次编号排列。OpenFlow规范中定义了流水线式的处理流程，如图5-8所示。当数据包进入Switch后，必须从FlowTable 0开始依次匹配；FlowTable可以按次序从小到大越级跳转，但不能从某一FlowTable向前跳转至编号更小的FlowTable。当数据包成功匹配一条规则后，将首先更新该规则对应的统计数据（如成功匹配数据包总数目和总字节数等），然后根据规则中的指令进行相应操作——比如跳转至后续某一FlowTable继续处理，修改或者立即执行该数据包对应的Action Set等。当数据包已经处于最后一个FlowTable时，其对应的Action Set中的所有Action将被执行，包括转发至某一端口，修改数据包某一字段，丢弃数据包等。OpenFlow规范中对目前所支持的Instructions和Actions进行了完整详细的说明和定义。

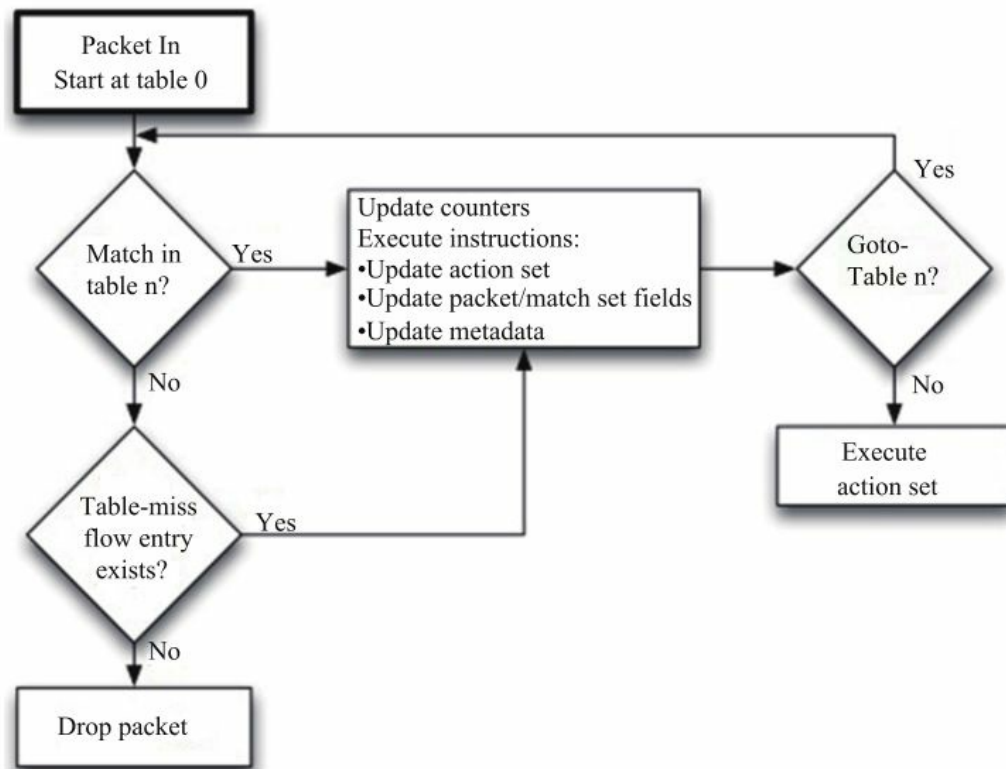


图5-8 OpenFlow流表匹配过程

另外，OpenFlow规范中还定义了很多其他功能和行为，比如OpenFlow对于QoS的支持（即MeterTable和Meter Bands的定义等），对于GroupTable的定义，以及规则的超时处理等。

（3）OpenFlow的通信通道

OpenFlow规范定义了一个OpenFlow Switch如何与Controller建立连接、通信以及相关消息类型等。

OpenFlow规范中定义了三种消息类型。

➤ Controller/Switch消息，是指由Controller发起、Switch接收并处理的消息，主要包括Features、Configuration、Modify-State、Read-State、Packet-out、Barrier和Role-Request等消息。这些消息主要由Controller用来对Switch进行状态查询和修改配置等操作。

➤ 异步（Asynchronous）消息，是由Switch发送给Controller、用来通知Switch上发生的某些异步事件的消息，主要包括Packet-in、

Flow-Removed、Port-status和Error等。例如，当某一条规则因为超时而被删除时，Switch将自动发送一条Flow-Removed消息通知Controller，以方便Controller做出相应的操作，如重新设置相关规则等。

➤ 对称（Symmetric）消息，顾名思义，这些都是双向对称的消息，主要用来建立连接、检测对方是否在线等，包括Hello、Echo和Experimenter三种消息。

图5-9 展示了OpenFlow和Switch之间一次典型的消息交换过程，出于安全和高可用性等方面的考虑，OpenFlow的规范还规定了如何为Controller和Switch之间的信道加密、如何建立多连接等（主连接和辅助连接）。

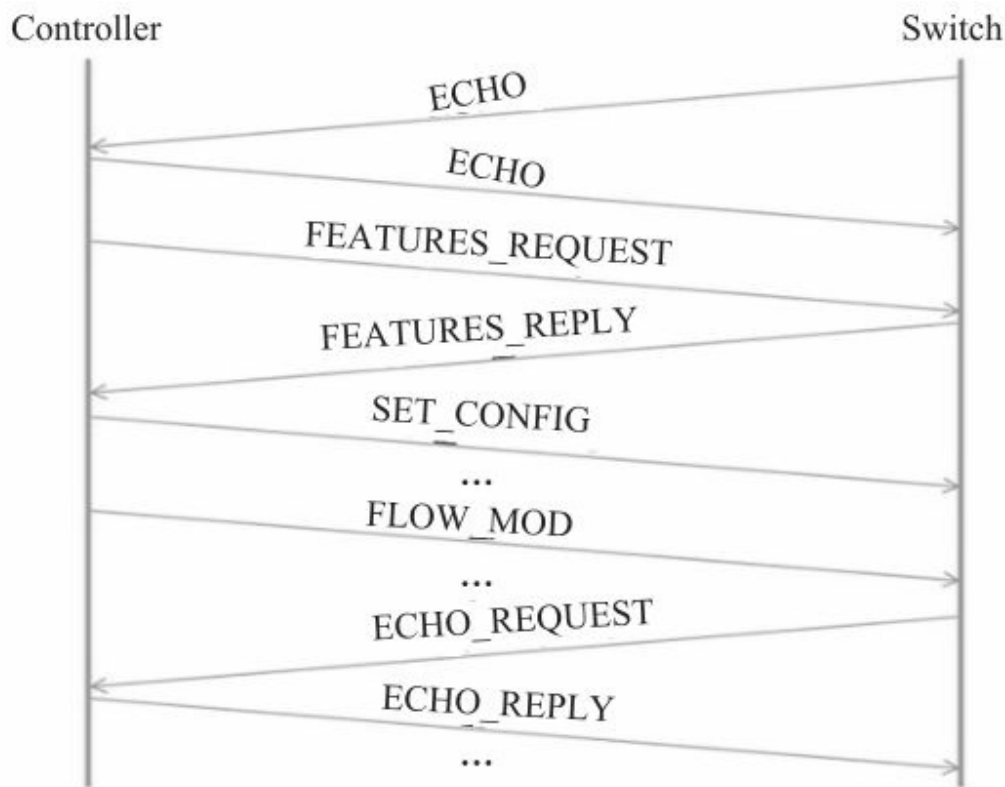


图5-9 控制器与Switch交互过程

（4）OpenFlow协议及相关数据结构

在OpenFlow规范的最后一部分，主要详细定义了各种OpenFlow消息的数据结构，包括OpenFlow消息的消息头等。

5.2.4 OF-Config

OF-Config是OpenFlow的伴侣协议。OpenFlow实现了Flow的match-action相关行为，但其他Flow依赖的资源都依赖OF-Config进行管理，包括：配置OpenFlow Controller地址、队列和物理端口、逻辑端口、通信信道、交换机能力发现等。对转发面的配置与管理，也有定义私有协议方案，如Open vSwitch使用自定义的OVSDB。

5.2.5 ONF及OpenDayLight标准联盟

开放网络基金会（ONF，Open Networking Foundation）是一个组织机构，致力于软件定义网络（SDN）的发展和标准化。ONF的主要任务是培养一个网络环境，这种环境能够支持OpenFlow，OpenFlow是一个允许服务器通知交换机往哪里发送数据包的协议。ONF的董事会成员包括微软、Google和Verizon。普通会员有几十个，包括思科、富士通、IBM、NEC、三星和惠普。

2013年4月，SDN行业组织OpenDaylight宣告成立。其成员包括Arista、Big Switch、博科、思科、思杰、戴尔、爱立信、富士通、IBM、英特尔、瞻博网络、微软、华为、NEC、Nuage Networks、PLUMgrid、红帽、VMware。在OpenDaylight项目中，网络行业将采取相同的方案研发他们的下一代技术，整体情况类似于大数据领域中利用Hadoop或带有WebKit的Web浏览器进行研发一样。OpenDaylight是一个研发实体，未来将与作为标准化实体的ONF形成互补。

OpenDaylight是一套以社区为主导的开源框架，旨在推动创新实施以及软件定义网络（简称SDN）透明化。面对SDN型网络，大家需要合适的工具帮助自己管理基础设施，这正是OpenDaylight的专长。作为项目核心，OpenDaylight拥有一套模块化、可插拔且极为灵活的控制器，这使其能够被部署在任何支持Java的平台之上。这款控制器中还包含一套模块合集，能够执行需要快速完成的网络任务（见图5-10）。

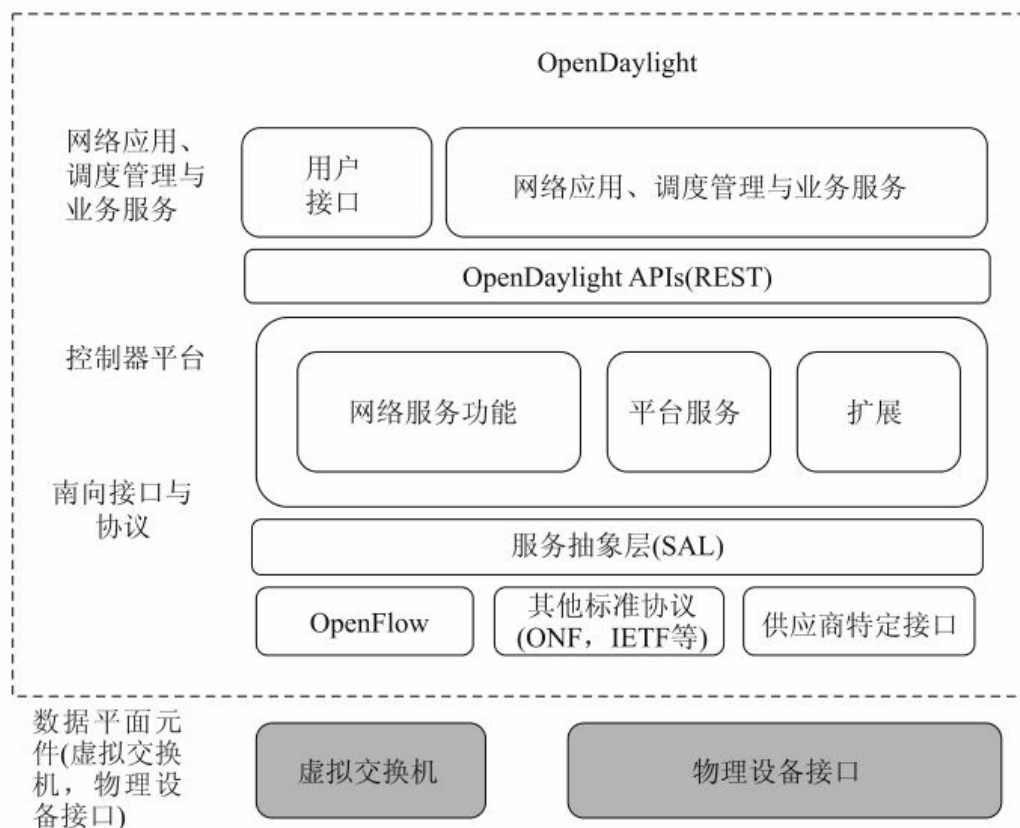


图5-10 OpenDaylight功能逻辑架构

OpenDaylight项目将研发一系列技术，包括在SDN中为网络设备提供集中控制的控制器。该控制器为云管理工具等具有网络感知功能的应用、交换机和其他网络中的硬件提供接口。该项目还将创建网络应用、网络虚拟化软件和其他组件。

如此众多的重量级厂商走到一起，说明SDN已成为一股不可忽视的潮流。不过，显而易见的问题是，种种不同的技术、见解如何真正聚合成一个切实可行的解决方案，面临的困难想来不少。因此，其发展前景尚待观察。这些重量级厂商汇聚在一起，一方面堵死了SDN新兴企业的发展壮大之路，另一方面也形成了和VMware生态系统抗衡的态势。

5.3 网络虚拟化关键技术：大二层实现

在数据中心网络中，“区域”对应于VLAN的划分。相同VLAN内的终端

属于同一广播域，具有一致的VLAN-ID，二层连通；不同VLAN内的终端需要通过网关互相访问，二层隔离，三层连通。传统的数据中心主要是依据功能进行区域划分，例如Web、APP、DB、办公区、业务区、内联区、外联区等。不同区域之间通过网关和安全设备互访，保证不同区域的可靠性、安全性。同时，不同区域由于具有不同的功能，因此需要相互访问数据时，只需终端之间能够通信，并不一定要求通信双方处于同一VLAN或二层网络。

传统数据中心服务器网络设计中，通常将二层网络的范围限制在网络接入层以下，避免出现大范围的二层广播域。虚拟机迁移技术可以使数据中心的计算资源得到灵活的调配，进一步提高虚拟机资源的利用率。但是虚拟机迁移要求虚拟机迁移前后的IP和MAC地址不变，这就需要虚拟机迁移前后的网络处于同一个层域内部。由于客户要求虚拟机迁移的范围越来越大，甚至是跨越不同地域、不同机房之间的迁移，这使得数据中心二层网络的范围越来越大，甚至出现了专业的大二层网络这一新领域专题。

所谓“大二层”是指所有VLAN都可以延展到所有汇聚层、接入层交换机的VLAN结构，这与传统数据中心VLAN往往终结在接入层交换机的做法不同。大二层网络结构的需求是由如下原因决定的。

- 服务器虚拟化的要求：所有主流服务器虚拟化技术都能够实现不同程度的虚拟机在线迁移，而虚拟机迁移前后其MAC/IP地址等不变，决定了其迁移的源和目的应在同一个VLAN。
- 网络业务整合的需要：新一代数据中心网络要求比传统网络具有更高的业务承载能力，各种应用（比如Oracle RAC等）等需要纯二层网络来提供其业务所需的低延迟、高吞吐、MAC层直接交换的网络环境。
- 智能业务整合、集中部署的需求：为面向范围更广的接入层提供智能服务资源池，智能服务被要求集中化部署，这就需要有智能服务要求的VLAN都能延展到智能服务设施所在的汇聚层。

随着应用的数量迅猛增加，二层网络的扩展会造成传统二层技术在链路冗余能力、负载均衡能力、可扩展性和网络稳定性上面的诸多问题。

5.3.1 CT流派：以交换机为中心的大二层技术

大二层首先需要解决的是数据中心内部的网络扩展问题，要通过大规模二层网络和VLAN延伸，实现虚拟机在数据中心内部的大范围迁移。由于数据中心内的大二层网络都要覆盖多个接入交换机和核心交换机，因此主要有以下两类技术。

基于跨交换机端口捆绑实现大二层网络

既然二层网络的核心是环路问题，而环路问题是随着冗余设备和链路产生的，那么如果将相互冗余的两台或多台设备、两条或多条链路合并成一台设备和一条链路，就可以回到之前的单设备、单链路情况，环路自然也就不存在了。尤其是交换机技术的发展，虚拟交换机从低端盒式设备到高端框式设备都已经广泛应用，具备了相当的成熟度和稳定度。因此，虚拟交换机技术成为目前应用最广的大二层解决方案。

虚拟交换机技术的代表是华为的Stack、H3C公司的IRF、Cisco公司的VSS。其特点是只需要交换机软件升级即可支持交换机虚拟机，应用软件成本低，部署简单。目前这些技术都是各厂商独立实现和完成的，只有同一厂商的相同系列产品之间才能实施虚拟化。同时，由于高端框式交换机的性能、密度越来越高，对虚拟交换机的技术要求也越来越高，目前框式交换机的虚拟化密度最高为4:1。虚拟交换机的密度限制了二层网络的规模大约在1万~2万台服务器左右。

基隧道技术

二层网络不能有环路，冗余链路必须要阻塞掉，但三层网络显然不存在这个问题，而且还可以做ECMP（等价链路）。通过在二层报文前插入额外的帧头，并且采用路由计算的方式控制整网数据的转发，不仅可以在冗余链路下防止广播风暴，而且可以做ECMP。这样可以在保持原有二层网络配置的简洁性的同时，将二层网络规模扩展到整张网络，而不会受核心交换机数量的限制（见图5-11）。

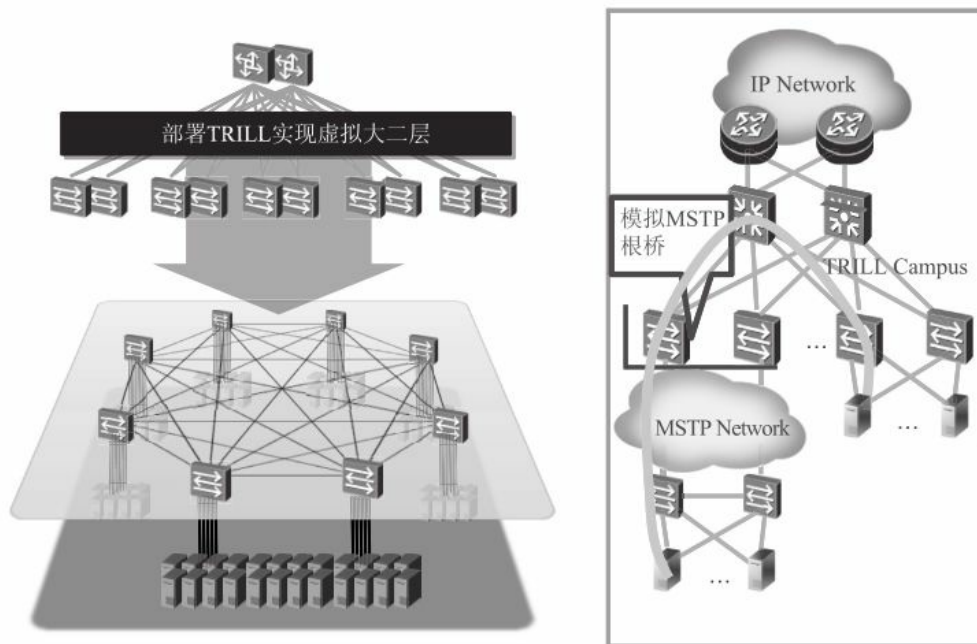


图5-11 Trill组网示意图

隧道技术的代表是TRILL、SPB，都是通过借用IS-IS路由协议的计算和转发模式，实现二层网络的大规模扩展。这些技术的特点是可以构建比虚拟交换机技术更大的超大规模二层网络（应用于大规模集群计算）。同时传统交换机不仅需要软件升级，还需要硬件支持。虽然TRILL成功扩展了虚拟机资源池的规模，但是需要在核心层上再增加一层设备来做网关。这导致网络结构变得复杂，管理难度增加，网络建设、运维成本都会增加。

以上基于各厂家私有的IRF/vPC等设备级的（网络N:1）虚拟化技术，虽然可以简化拓扑、具备高可靠性的能力，但是对于网络有强制的拓扑形状限制，在网络的规模和灵活性上有所欠缺，只适合小规模网络构建，且一般适用于数据中心内部网络。而为了大规模网络扩展而产生的TRILL/SPB/FabricPath/VPLS等技术，虽然解决了上述技术的不足，但对网络有特殊要求，即网络中的设备均要软硬件升级而支持此类新技术，这将带来部署成本的上升，并且还存在厂家互通问题，多数用在新建数据中心场景。

5.3.2 IT流派：以服务器叠加网为中心的Overlay技术

第二类是IT的方案——软件Overlay，将二层网络封装后在三层范围内进行扩展的VxLAN方案。其可以很好地解决虚拟机灵活迁移、超大规模

（16M）VLAN的问题。

以服务器Overlay叠加网为中心的IT流派代表是Google和微软。

Overlay在网络技术领域，指的是一种网络架构上叠加的虚拟化技术模式，其大体框架是对基础网络不进行大规模修改的条件下，实现应用在网络上的承载，并能与其他网络业务分离，并且以基于IP的基础网络技术为主。Overlay采用全新方式的解决以下问题。

（1）虚拟机迁移范围受到网络架构限制

Overlay是一种封装在IP报文之上的新的数据格式，因此这种数据可以通过路由的方式在网络中分发，而路由网络本身并无特殊网络结构限制，具备良性大规模扩展能力，并且对设备本身无特殊要求，以高性能路由转发为佳，且路由网络本身具备很强的故障自愈能力、负载均衡能力。采用Overlay技术后，企业部署的现有网络便可用于支撑新的云计算业务，改造难度极低（除性能可能是考量因素外，技术上对于承载网络并无新的要求）。

（2）虚拟机规模受网络规格限制

虚拟机数据封装在IP数据包中后，对网络只表现为封装后的网络参数，即隧道端点的地址，因此，对于承载网络（特别是接入交换机），MAC地址规格需求极大降低，最低规格也就是几十个（每个端口一台物理服务器的隧道端点MAC）。

（3）网络隔离/分离能力限制

针对VLAN数量4000以内的限制，在Overlay技术中引入了类似12比特VLAN ID的用户标识，支持千万级以上的用户标识，并且在Overlay中沿袭了云计算“租户”的概念，称之为Tenant ID（租户标识），用24或64比特表示。针对VLAN技术下网络的TRUNK ALL（VLAN穿透所有设备）的问题，Overlay对网络的VLAN配置无要求，可以避免网络本身的无效流量带宽浪费，同时Overlay的二层连通基于虚拟主机业务需求创建，在云的环境中全局可控。

IETF在Overlay技术领域有如下三大技术路线。

(1) VxLAN

VxLAN是将以太网报文封装在UDP传输层上的一种隧道转发模式，目的UDP端口号为4798；为了使VxLAN充分利用承载网络路由的均衡性，VxLAN将原始以太网数据头（MAC、IP、四层端口号等）的HASH值作为UDP的号；采用24比特标识二层网络分段，称为VNI（VxLAN Network Identifier），类似于VLAN ID作用；未知目的、广播、组播等网络流量均被封装为组播转发，物理网络要求支持任意源组播（ASM）。

(2) NVGRE

NVGRE是将以太网报文封装在GRE内的一种隧道转发模式；采用24比特标识二层网络分段，称为VSI（Virtual Subnet Identifier），类似于VLAN ID作用；为了使NVGRE利用承载网络路由的均衡性，NVGRE在GRE扩展字段flow ID，这就要求物理网络能够识别出GRE隧道的扩展信息，并以flow ID进行流量分担；未知目的、广播、组播等网络流量均被封装为组播转发。

(3) STT

STT利用了TCP的数据封装形式，但改造了TCP的传输机制，数据传输不遵循TCP状态机，而是全新定义的无状态机制，将TCP各字段意义重新定义，无需三次握手建立TCP连接，因此称为无状态TCP；以太网数据封装在无状态TCP；采用64比特Context ID标识二层网络分段；为了使STT充分利用承载网络路由的均衡性，将原始以太网数据头（MAC、IP、四层端口号等）的HASH值作为无状态TCP的源端口号；未知目的、广播、组播等网络流量均被封装为组播转发。

这三种二层Overlay技术，大体思路均是将以太网报文承载到某种隧道层面，差异性在于选择和构造隧道的不同，而底层均是IP转发。VxLAN和STT对于现网设备对流量均衡要求较低，即负载链路负载分担适应性好，一般的网络设备都能对L2~L4的数据内容参数进行链路聚合或等价路由的流量均衡，而NVGRE则需要网络设备对GRE扩展头感知，并对flow ID进行HASH，并且需要硬件升级；STT对于TCP有较大修改，隧道模式接近UDP性质，隧道构造技术属于革新性，且复杂度较高，而VxLAN利用了现有通用的UDP传输，成熟性极高。总体比较，VxLAN技术相对具有优势。

目前的虚拟化主机软件在vSwitch内支持VxLAN，使用VTEP（VxLAN Tunnel End Point）封装和终结VxLAN的隧道。为了更加简化VxLAN Overlay网络的运行管理，便于云的服务提供，各厂家使用集中控制的模型，将分散在多个物理服务器上的vSwitch构成一个大型的、虚拟化的分布式Overlay vSwitch。只要在分布式vSwitch范围内，虚拟机在不同物理服务器上迁移，便被视为在一个虚拟的设备上迁移，如此大大降低了云中资源的调度难度和复杂度。

为了解决采用Overlay隧道后的广播报文（ARP、DHCP等）和单播MAC地址位置学习问题，业界采用了一些创新的技术手段。例如在开源OpenStack中的ML2 Plugin采用12-Population机制在虚拟机创建时预下发本子网的所有虚拟机MAC地址和隧道端点信息。也可以采用SDN控制器来集中代理ARP/DHCP报文，按需下发隧道通信使用的流表。

5.4 网络虚拟化关键技术：多租户网络实现

多租户的计算模式是云计算技术架构中面向服务的最为典型的应用模式。它要求服务器计算环境，存储资源及其网络资源的设计和部署必须满足自动化、快速性、动态性、移动性、安全性和面向商业服务等需求。按照云计算资源虚拟化的技术要求，对网络拓扑和链路进行虚拟化，并对网络资源实现按照策略的隔离和共享（见图5-12）。

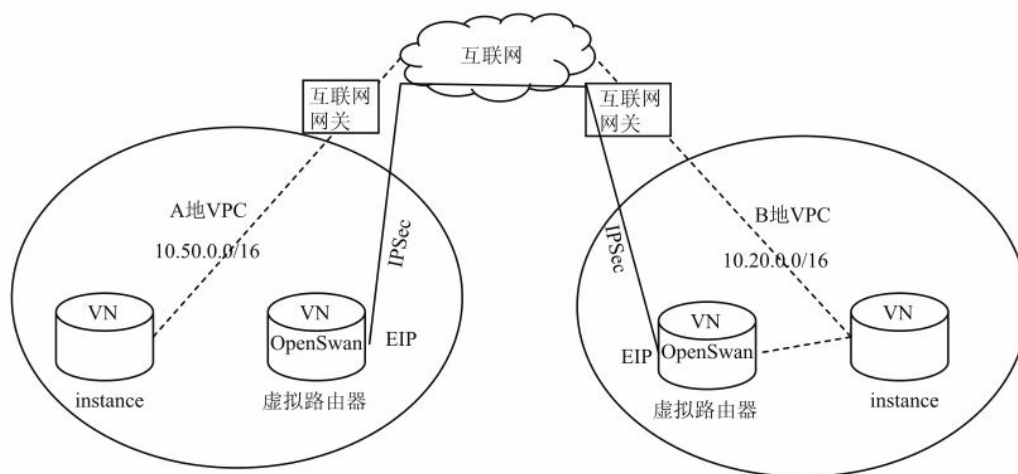


图5-12 多租户网络

多租户网络必须满足虚拟机之间的隔离需求

每个租户拥有虚拟服务器资源池中的一个虚拟服务器或一组虚拟服务器。同时每个租户都必须有自己的独立的网络链路，租户独立的网络资源包括以下几个方面。

➤ 独立的二层广播域

传统网络通常使用VLAN进行隔离，为每个租户提供一组VLAN资源，但是4K的VLAN个数限制了网络规模。对于仅使用独立云主机，对网络其他资源无特别要求（例如IP地址重叠）的应用，可以采用ARP代答的方式消除VLAN网络中不同租户的广播。同时支持通过租户自定义的安全组（一组ACL规则）实现不同租户虚拟机间的4层安全隔离；可采用VxLAN/ NVGRE/STT等隧道技术来突破VLAN规格限制。

➤ 独立的IP地址空间

通常的做法是为每个租户的网络分配独立的虚拟DHCP服务器来实现重叠的IP地址分配。

➤ 独立的虚拟路由器与防火墙

当有了独立的地址空间后，对于三层以上网络，需为每个租户网络分配独立的虚拟路由器和防火墙。使用传统网络设备如物理交换机、物理防火墙，在转发面可使用VRF（Virtual Routing Forwarding）技术实现一虚多。物理网络设备虚拟化存在配置复杂、数据流量迂回等缺点。于是产生了对网络设备NFV（Network Function Virtualization）的需求，对于路由器、防火墙、LB等网络节点，采用x86虚拟机或裸机实现，例如软件Vyatta路由器、VmwareEdge、Cisco VSG等。

与硬件网络设备相比，软件NFV存在转发性能和时延方面的先天不足。IT流派采用分布式与集群技术来解决单个虚拟化网络设备性能瓶颈问题。分布式路由器（Distributed Virtual Router）技术可将路由器网管逻辑部署在每个服务器主机节点，采用本地或集中ARP代答的方式支持ARP报文。可以在分布式路由器中动态或静态添加主机路由，分布式路由器实现东西向报文的路由与卸载。对于南北报文，可支持多台路由器作为集群，采用SDN控制器根据流量负载和设备状态动态为虚拟机选择

出口路由器。分布式防火墙（Distribute Firewall）将在每个物理主机Hypervisor层部署防火墙功能，采用NameSpace技术进行流表隔离，将安全过滤分布在每个节点，同时通过集中的安全策略简化管理负责度。弹性负载均衡（Elastic Load Balance）技术可感知网络负载情况，并根据负载状况，动态申请负载均衡虚拟机，与DNS结合实现弹性伸缩。由于不是每个NFV设备都可支持Overlay隧道技术，通常将隧道终结在vSwitch上，NFV设备与Overlay网络的对接需要通过VLAN到VxLAN Mapping来实现。

多租户的网络必须是策略驱动的网络

每个租户必须和云服务提供商签署网络服务策略，即每个租户有自己独立的网络策略部署方案。云计算系统如OpenStack、CloudStack均实现对网络资源的多租户标识。通过开放的API接口，租户可使用云计算系统提供的GUI界面或调用底层API实现自助的管理网络策略与配置，包括申请子网、定义IP地址、申请路由器、申请弹性IP（公有地址与虚拟机私有地址的NAT映射）、创建安全组、定义负载均衡规则、定义防火墙规则、定义VPN规则等。

多租户的网络必须确保每个租户不同的服务质量

在多租户网络中，每个租户有不同的应用，所以云服务提供商必须确保每个租户的服务质量，即服务的SLA。如对带宽的保证，在流量拥塞的时候确保最低带宽，避免拥塞等。所以在多租户网络中，每个用户的QoS策略及其流量策略非常重要。多租户的虚拟网络接口可按照层次关系进行如下的设计：vNIC（虚拟机）→vPort（虚拟交换机）→pNIC（网卡）→VIF（交换机/Router）→VIF（Firwall/LB）。其通常在如下节点定义Qos策略：

- 在云主机Hypervisor层定义vNIC收/发方向的Car、Shaping，同时支持基于端口或流对报文进行DSCP标签；
- 考虑通过独立的网卡或pNIC保证管理、存储平面带宽需求；
- 在数据中心出口，对租户的虚拟防火墙、弹性IP、负载均衡的带宽进行限制。

多租户网络必须能够实现租户跨广域网或城域网的互联互通

在多租户网络中，租户的虚拟服务器的部署并不一定都集中在一个数据中心，而是大部分分布在不同的数据中心，有的需要跨越城域网甚至广域网。在这种情况下，多租户网络必须能够跨越互联网实现租户的互联互通。例如在不同数据中心间可采用NVGRE/ VxLAN隧道实现跨数据中心的二层互通；在云数据中心与租户网络间可采用IPSec-VPN、SSL-VPN、L2TP-VPN实现互通。

5.5 网络虚拟化端到端解决方案

SDN提供了智能、集中控制的全局优化，应用驱动，开放可编程，端到端QoS，网络快速修复能力。随着OpenFlow/SDN概念的发展和推广，其研究和应用领域也得到了不断拓展。下面将举几个典型的研究案例来展示OpenFlow的应用。

基于业务和应用驱动的SDN网络管理优化

SDN/OpenFlow提供了一个简化的网络自动化配置流程。系统具备统一标准的数据库，所有设备提供基于OpenFlow和Restful的接口。使用标准化的API提供自动化的服务创建过程，从而消除了使用CLI和其他人工服务的创建过程。标准OpenFlow也消除了需要分别使用厂商定制的EMS来逐个配置各个网络服务的复杂过程。

网络虚拟化（Slicing/Traffic Isolation）

网络虚拟化需要能抽象出底层网络的物理拓扑，在逻辑上对网络资源进行分片或者整合，从而满足各种应用对于网络的不同需求。虚拟网络也可支持IP地址重叠，当前是采用VRF进行隔离，但是VRF难于配置与部署。为了达到网络分片的目的，通过OpenFlow Controller，其可以看做是其他不同用户或应用的Controllers与网络设备之间的一层代理。因此，不同用户或应用可以使用自己的Controllers来定义不同的网络拓扑，同时又可以保证这些Controllers之间能够互相隔离而互不影响。

分布式的负载均衡

传统的负载均衡方案一般需要在服务器集群的入口处，通过一个 gateway 或者 router 来监测、统计服务器工作负载，并据此动态分配用户请求到负载相对较轻的服务器上。既然网络中所有的网络设备都可以通过 OpenFlow 进行集中式的控制和管理，同时应用服务器的负载可以及时地反馈到 OpenFlow 的 Controller，Controller 就可以根据这些实时的负载信息，重新定义网络设备上的 OpenFlow 规则，从而将用户请求（即网络包）按照服务器的能力进行调整和分发。

绿色节能的网络服务

在数据中心和云计算环境中，如何降低运营成本是一个重要的研究课题。SDN 可根据工作负荷按需分配、动态规划，不仅可以提高资源的利用率，还可以在网络负载不高的情况下选择性地关闭或者挂起部分网络设备，使其进入节电模式，达到节能环保、降低运营成本的目的。

动态插入安全与策略

使用 SDN 实现自动流量导流，从而提供防火墙和入侵检测系统（IDS）服务。当前大多数的安全策略受到 VLAN 或接口的限制，并且使用静态的配置，无法感知具体应用上下文信息。尽管可通过使用 802.1.x 进行动态策略与身份标识管理的增强，但还是无法提供足够灵活的安全策略管理能力。在 SDN 环境中，SDN Controller 可理解流的上下文信息（用户、时间、应用和其他外部参数），让管理员可以配置更好颗粒度的策略，并应用到交换中。

SDN 网络中，可由基于 OpenFlow 的路由功能，将流量引导到 IP 服务（如防火墙和 IDS 中）。SDN 控制器和 IP 服务软件向路由器提供路由和配置命令。可通过 SDN 智能化路由，减少流入 IP 服务应用程序的流量，从而提高网络的利用率，降低 I/O 端口的消耗。

广域网络虚拟化

在保护现有的 L2/L3 VPN-IP 网络的同时，提供了一个 OpenFlow 叠加到跨数据中心网络连接中。

在没有实时全网信息时，按照局部 FIB 信息的最短路径会导致大量数据被引导到某些路径，导致网络在某些地段阻塞。基于 OpenFlow 的 Overlay 网络和广域网 SDN 控制器被添加到现有的生产网络。OpenFlow

并不影响传统的流量和基于硬件提供的防护能力。使用Hybrid端口模式，降低初始部署风险，以持续逐步增加OpenFlow覆盖服务。由于SDN处理不断增加的流量，共享网络容量的增加会减少。SDN提供更好的网络使用情况的可见性和路由的灵活性，以达到更高的利用率水平，例如Google在使用SDN/Openflow以前，网络的使用率在30%左右，而采用SDN/Openflow后，网络利用率提高到90%以上。

5.6 网络云化还有多远

与计算虚拟化相比，网络虚拟化以及网络云化的历史并不算久，世界各地大规模商用的案例比较少。但现阶段小规模，以虚拟交换机、分布式虚拟交换机、虚拟路由器、虚拟防火墙、虚拟负载均衡器为代表的，运行在私有云内的网络虚拟化商用案例还是比较多的。商用网络虚拟化案例的最大特点是，对现有网络硬件设备基本没有改动，而是通过服务器虚拟化层运行的虚拟化与云计算软件完成网络虚拟化功能。这种数据中心私有云内部的网络虚拟化会随着企业私有云建设而变得普遍化，由于对现有网络硬件要求不高，其推广阻力也很小。

大部分企业数据中心规模并不大，网络结构也不是很复杂，网络设备规模庞大且结构复杂的当数电信运营商了（还有少量超大型企业，特别是超大型互联网企业）。网络何时走向彻底云化，或者说网络云何时普遍商用，主要要看电信运营商、网络设备供应商、IT系统供应商之间的互动博弈。电信运营商无疑希望大幅降低网络设备采购成本与运营管理复杂度，这样可提升自动化与弹性的配置管理能力，增强赢利能力，而网络设备供应商期望赚取更多的利润。供需双方本身就存在这个矛盾。谁有网络控制权谁就有话语权，而网络云化的一个重要手段就是让网络设备把控制权交出来（交给IT系统），传统网络设备供应商肯定是不情愿的。但不情愿也没有太好的办法，网络设备是可以替换的，而且厂商众多，为了生存，只能全力竞争。运营商一旦认准基于云计算的开放网络架构这个大方向，天平是会向需方倾斜的。只要运营商从战略运营上开始有动作，网络云化的时间表就会被排得非常紧密，短则一两年，长则不超过三五年。运营商的网络云化，实际意味着全球骨干网络云化，那么作为网络的末梢节点——企业内部数据中心网络，则会跟随这个大趋势快速云化。

网络云化，可能改变不了传统电信运营商“数据物流商”的角色，但可以大幅压低“数据物流”的成本。这与实体物流行业很类似，10年前最头疼

的事情是网购省下的钱都被快递费用占去了，而今天不仅很多网购免运费了，而且由7天送达，缩短到24小时、4个小时、2个小时，甚至送达时间以分钟计算。未来（可能5~10年），电信运营商可能只需要向租用网络宽带出口的企业（包含互联网企业）单向收费，而面向普通个人则免费（或准免费）泛在接入，即可实现产业繁荣（以运营商间能够实现充分竞争为前提）。

第6章 面向企业关键应用性能提升和存储管理简化的存储虚拟化

6.1 云计算的存储虚拟化概述

企业级存储中使用的传统SAN、NAS设备，在云计算中面临了很多的问题，主要问题如下。

（1）存储弹性问题

企业级存储无法满足多业务不同负载、动态的资源变化需求，在不同租户和不同应用对资源有不同要求的时候，很难方便地做出调整，包括性能和容量资源的弹性调配，而云计算中多租户多业务负载下资源的弹性是极其重要的核心要素。

（2）存储扩展问题

传统存储的扩展性面临了多个瓶颈，如机头、前后端网络、磁盘与CPU/MEM资源不同步扩展等，都是传统存储无法做到线性扩展的几个关键因素。

（3）形态和实施的成本、复杂性问题

传统存储在部署的时候，需要独立的存储网络，用于多主机互联，特别是针对性能较高的FC网络，在实施的时候成本较高、组网实施复杂，不利于大规模集群的简化部署实施。

（4）大规模集群下的容错和可靠性问题

在规模很大的云计算环境下，需要具备跨机房、跨机柜、跨服务器的数据保护机制，即使在机柜故障等场景下，数据仍然不丢失，仍然可访问。

（5）灵活的软件定义策略问题

在云计算环境下，不同的租户、不同的业务应用对存储有着不同的要求，需要底层存储具备灵活的软件定义策略支持，允许用户按需进行存储的策略配置（如定义多大的容量、多少IOPS、多大的SSD Cache缓存、什么样的数据冗余和可靠性要求等），底层存储可以根据这些软件定义策略进行资源的调配，按需自动地满足上层业务和应用的需求。

云计算的存储虚拟化概述如图6-1所示，其中包含了传统存储的虚拟化、分布式存储的池化和加速以及软件定义的存储策略控制三个部分。

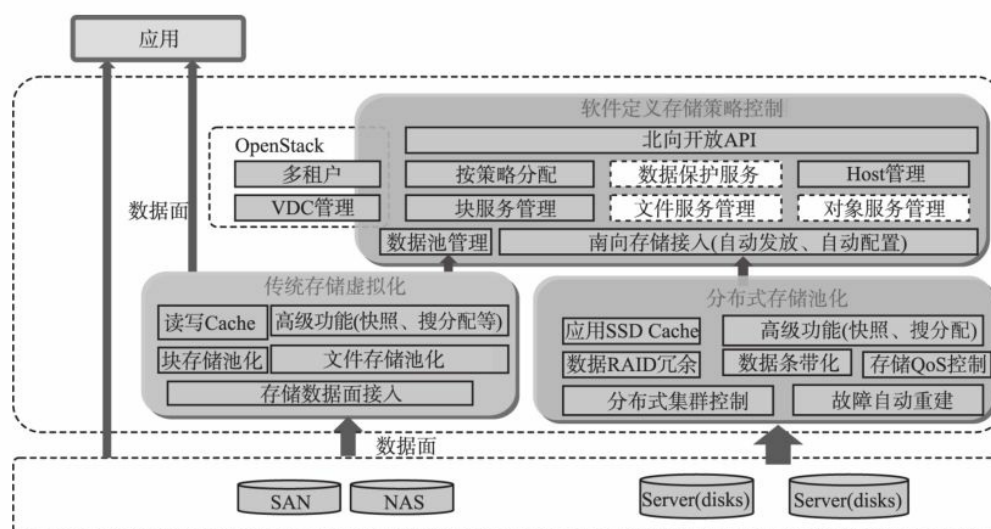


图6-1 存储虚拟化框架

6.2 灵活的软件定义存储

虚拟化、云计算时代，应用对于存储有了新的需求：

- 希望存储系统能够简化管理，兼容不同的存储设备，快速获取存储资源；
- 能够基于策略驱动，利用虚拟机粒度的存储策略，保证服务质量（QoS）；
- 希望新的存储方案具有高性价比，提供低成本、高性能的存储解决方案；

- 存储系统需要具备高可扩展性，实现系统规模动态扩展。

传统存储虚拟化技术有各自的缺陷：基于网络的存储虚拟化技术增加了硬件成本，系统扩展性差，存在性能损耗；基于存储设备的虚拟化技术的异构兼容能力弱，快照、瘦分配有较大的损耗性能；基于主机的存储虚拟化技术一般采用主机侧SSD缓存加速技术，但这个技术与阵列联动困难，快照回滚存在一致性问题，没有虚拟化能力；而且这三种技术都无法实现细粒度的策略。在这种情况下，一种新的存储虚拟化技术——软件定义存储的解决方案应运而生。

软件定义存储的需求模型如图6-2所示。

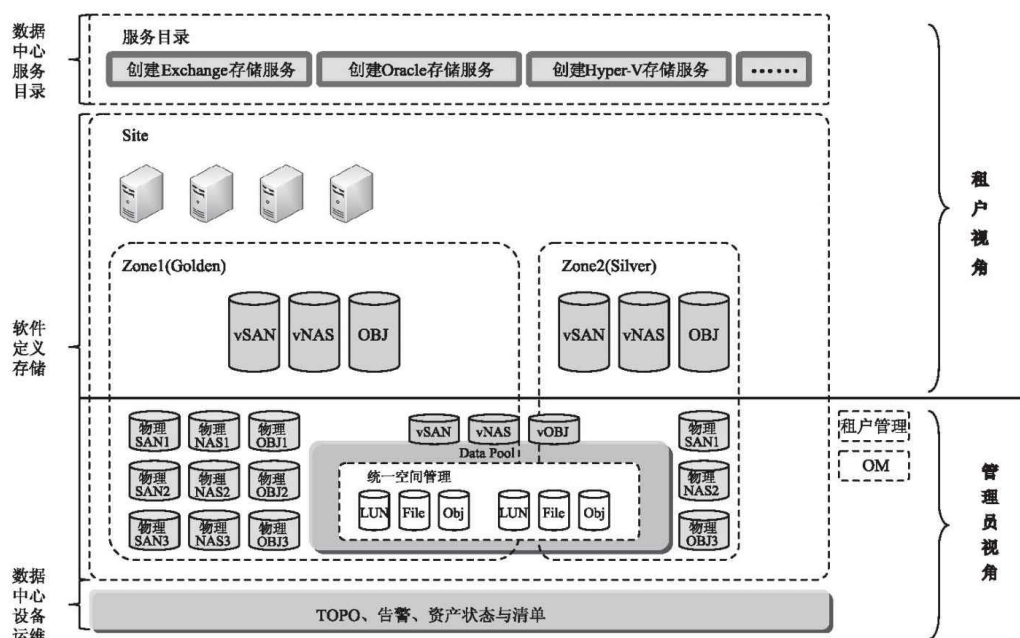


图6-2 软件定义存储需求模型

从租户视角来看，其典型操作包括：

- 根据服务目录操作高级服务；
- 查询卷、文件、对象存储的容量；
- 卷、文件、对象存储的增删改查；

- 主机挂载、卸载卷和文件目录。

从管理员视角来看，其典型操作包括：

- 查询TOPO接连关系；
- 分配物理设备到租户视角的Cell以及Zone；
- 分配物理空间给数据池；
- 存储设备与数据池OM；
- 主机安装Agent；
- 服务目录管理；
- 租户权限管理。

在功能上，软件定义存储对租户屏蔽物理存储设备、对管理员提供将物理设备映射到逻辑概念的手段。在架构上，对上提供补充存储类服务操作到数据中心服务目录，对下集成设备管理信息。

IDC给出软件定义存储（SDS）的定义：可以安装在商用资源（x86硬件、虚拟机监控程序或者云）和/或现有计算硬件上的任何存储软件堆栈。此外，为了取得资格，基于软件的存储堆栈应该提供一套完整的存储服务，以及在基础的持续数据配置资源之间的联邦，这使其租户的数据可以在这些资源之间流动。

软件定义存储有以下价值。

- 完全通过软件实现存储的高级特性：快照、克隆、瘦分配、高速缓存等都不依赖于存储设备。
- 策略驱动的设计：传统存储无法与应用配合，性能低下，基于策略的存储，能够为不同的应用提供不同的QoS。
- 简化存储管理：可以实现一次配置，多次使用，计算管理员只

需在系统初始配置或者扩容时需要存储管理员的参与，后续应用需要存储资源时，能够做到即时分配。

➤ 高性价比

- 性能：Cache+低端存储>高端存储。
- 成本：Cache+低端存储<高端存储。
- 可以利旧。

6.3 传统存储SAN/NAS的虚拟化

当仅需要单个主机服务器（或单个集群）访问多个磁盘阵列时，可以使用基于主机的存储虚拟化技术。该技术又称为逻辑卷管理，通常由主机操作系统下的逻辑卷管理软件实现。逻辑卷管理软件把多个不同的物理磁盘映射成一个虚拟的逻辑块空间。当存储需求增加时，逻辑管理软件能把部分逻辑空间映射到新增的磁盘阵列，因此可以在不中断运行的情况下增加或减少物理存储设备。

基于主机的存储虚拟化示意图如图6-3所示。

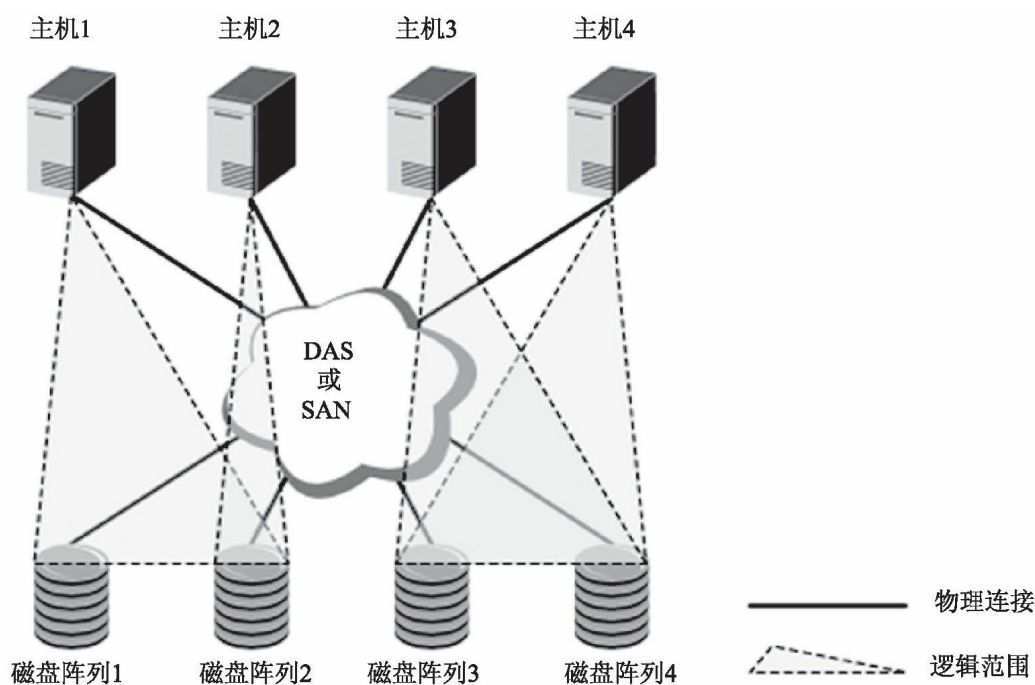


图6-3 基于主机的存储虚拟化

主机1可以使用磁盘阵列1和阵列2上的存储空间，主机2可以使用磁盘阵列2上的存储空间，主机3和主机4均可使用磁盘阵列3和阵列4上的存储空间。

该技术使主机经过虚拟化的存储空间跨越多个异构的磁盘阵列，因此常用于在不同磁盘阵列之间做数据镜像保护。

该技术的优点：

- 支持异构的存储系统；
- 容易实现，不需要额外的特殊硬件；
- 开销低，不需要硬件支持，不修改现有系统架构。

该技术的缺点：

- 占用主机资源，降低应用性能；
- 存在操作系统和应用的兼容性问题；
- 导致主机升级、维护、扩展复杂，容易造成系统不稳定；
- 需要复杂的数据迁移过程，影响业务连续性。

如果仅针对传统中低端存储设备整合的软件定义存储方案，其架构要简单很多，架构框架如图6-4所示。

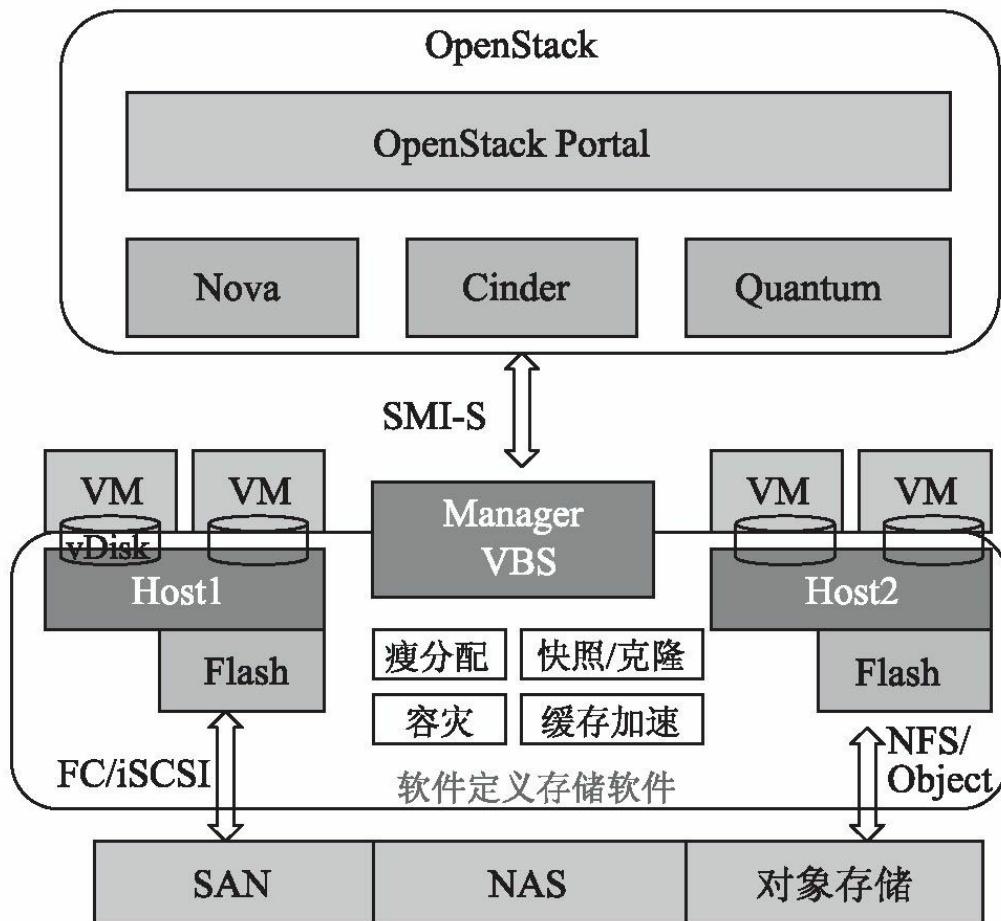


图6-4 中低端存储整合软件定义存储系统框架

中低端存储的存储性能差，LUN数量有限，无NVRAM、SSD加速，无快照/克隆，瘦分配等功能，可通过主机侧的Flash硬件和软件定义存储软件能力，整合现有中低端SAN、NAS等存储资源，提供性能高、功能强的软件定义存储解决方案。

面向中低端存储整合的软件定义存储系统主要是一套存储虚拟化软件，运行在Host OS上，它具有强大的异构能力，底层能够兼容块、文件或者对象。软件定义存储系统具备线性扩展能力，具有高速的分布式Flash Cache，能够实现性能无损的快照和瘦分配，能够实现VM粒度的策略驱动，拥有丰富的对外接口，能够对外提供块、文件或对象接口。系统主要提供卷管理服务、I/O服务、元数据服务。各种服务可以融合部署，也可分离部署。软件定义存储的软件逻辑如图6-5所示。

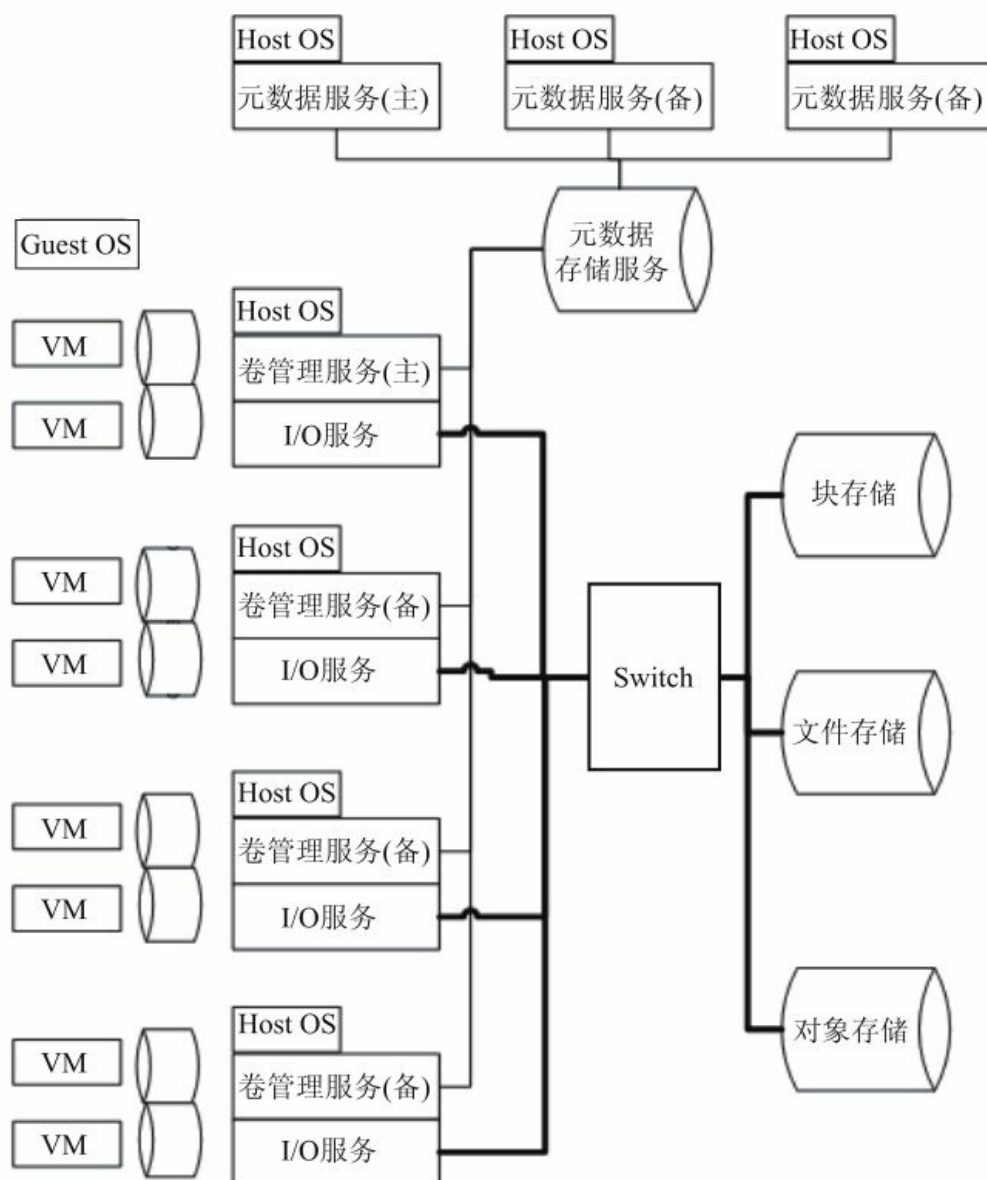


图6-5 面向中低端存储设备软件定义存储的软件逻辑

6.4 分布式存储池化和加速

6.4.1 分布式存储概述

随着企业面临的竞争环境越来越激烈、新业务上线时间要求越来越短，其IT系统需要从传统的成本中心转变为提升企业竞争力的利器，帮助企业提升竞争力并实现商业成功。作为存放企业数据资产的存储系统，不但要满足业务所需要的高性能、高可靠等基本诉求，更要满足未来业务

的发展、提升业务的敏捷性，帮助业务更快更好地适应竞争环境的需要。

计算与存储在过去20年一直在非均衡地发展。摩尔定律设想单位面积晶体管数量每18月增加1倍，对应单位价格的计算性能将翻2倍以上。回顾过去20年：处理器和网络带宽分别提升了3000倍和1000倍，但磁盘和内存带宽仅提升120倍，远落后于摩尔定律。阿姆达尔定律认为，计算系统对某一部件采用更快执行方式所能获得的系统性能改进程度，取决于这种执行方式被使用的频率，或所占总执行时间的比例。对于多数应用而言，基本均属于CPU计算与内/外部存储（Mem/Disk）的串联模型。换言之，系统中最慢部分（存储）的效率将决定和制约整个系统的效率。

在云计算集中化数据中心资源池环境中，由于GE以太网络的延伸作用，远端RAM/ SSD的容量与本地存储相差不超过1个数量级，数据访问时延则比本地HDD减少30倍，带宽降低1倍。

从20世纪80年代到21世纪的前10年，计算与存储经历了一次分离的变革。这次分离是由计算与存储的性能发展差距导致的。基于晶体管的计算与基于机械硬盘介质的处理性能差距越来越大，以及存储数据的重要性不断增加，为便于提升资源利用率，终于导致了计算、存储的架构分离，各自进行资源最优配置（见图6-6）。

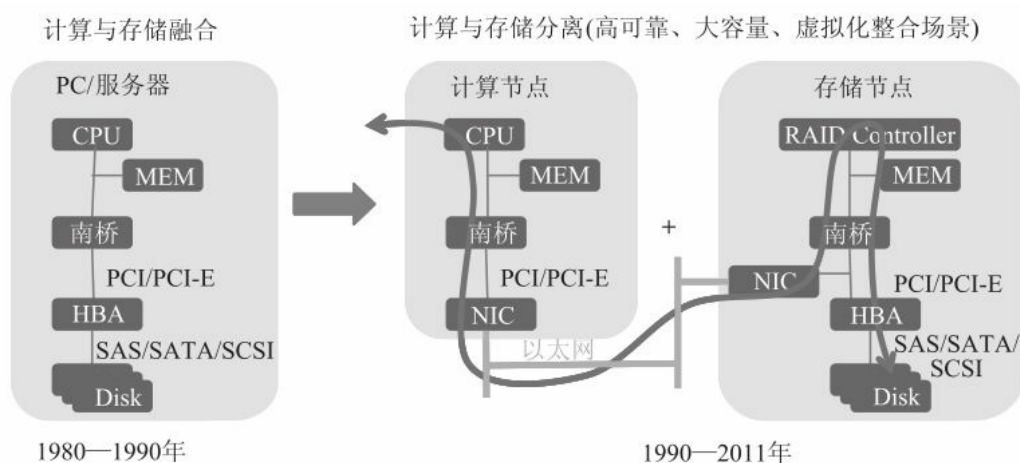


图6-6 存储由合到分的历史

下面我们介绍发生在世纪之交的这次计算、存储分离带来的其他优势。

- 应用计算的高可靠HA迁移能力：通过多个计算实例共享相同存储，在计算节点发生故障后，无需耗时的外存储数据迁移，即可快速在其他计算节点上恢复故障应用。
- 计算处理逻辑与应用数据分离，使得计算节点的更新（软硬件升级）不影响数据的可获得性。
- 计算与存储各自资源利用率最大化及技术独立发展：计算与存储各自独立组成水平资源池，存储资源池按需向计算应用实例供给存储资源。

随着企业业务的不断发展，特别是互联网突飞猛进的发展，这种计算/存储分离架构面临的挑战和问题越来越突出，具体表现在如下几个方面。

- 计算与存储物理架构上的人为分割，导致系统成本居高不下：目前业界主流的磁盘阵列软硬件普遍价格昂贵，且磁盘阵列自身的策略管理配置非常复杂，维护成本居高不下，尤其是在缺乏IT专业人员的场景（如行业分支机构、SME等）下，由人工误操作导致的业务中断，其风险较高。
- SAN机头成为可能制约系统扩展性的单点瓶颈：随着计算集群规模的不断扩展，由于存储资源池集中式控制机头的存在，使得共享存储系统的可扩展性及可靠性受到制约；如采用多个SAN系统，则会导致各独立SAN系统之间存储无法共享。
- SSD存储介质的引入，使得SAN控制机头可能成为系统性能瓶颈，SSD与归属计算节点CPU/MEM之间的最高效连接方式应为PCI-E，如果将SSD按照传统存储模式集中部署，则控制机头有可能成为CPU与SSD间高吞吐I/O带宽的瓶颈，并且系统复杂度更高。
- 集中式SAN控制机头可能成为影响系统整体可靠性的单点故障风险点：在虚拟化服务器整合环境下，成百上千VM共享同一存储资源池，一旦磁盘阵列控制器发生故障，将导致整体存储资源池不可用。尽管SAN控制机头自身具有主备机制，但依然存在异常条件下主备同时故障的可能性。

➤ 集群组网环境下，各计算节点的内存/SSD作为分层存储的缓存彼此孤立，只能依赖集中存储机头内的缓存实现I/O加速：共享存储的集群内各节点Cache容量有限，但不同节点Cache无法协同，且存在可靠性问题，导致本可作为集群共享缓存资源的容量被白白浪费。

➤ 虚拟化技术迅猛发展，虚拟机技术给服务器带来更高的利用率、给业务带来更便捷的部署，降低了TCO，因而在众多行业得到了广泛的应用。与此同时，虚拟机应用给存储带来以下挑战：第一，相比传统的物理服务器方式，单个存储系统承载了更多的业务，存储系统需要更强劲的性能来支撑；第二，采用共享存储方式部署虚拟机，单个卷上可能承载几十或上百的虚拟机，导致卷I/O呈现更多的随机特征，这对传统的Cache技术提出挑战；第三，单个卷承载多个虚拟机业务，要求存储系统可协调虚拟机访问竞争，保证对QoS要求高的虚拟机获取到资源实现性能目标；第四，单个卷上承载较多的虚拟机，需要卷具有很高的I/O性能，这对传统受限于固定硬盘的RAID技术提出挑战；第五，虚拟机的广泛使用，需要更加高效的技术来提高虚拟机的部署效率，加快新业务的上线时间。

面对这些挑战，正所谓“合久必分、分久必合”的哲学规律在IT领域同样上演。针对大多数企业事务型IT应用而言，关注核心在于信息数据的“即时处理”而非“存储/归档”，因此存储向计算的融合再次符合企业业务应用的根本诉求。通过引入Scale-out存储机制，可实现服务器集群环境下DAS直连硬盘的资源池化和虚拟化，推动计算与存储从“物理分离”架构向“物理”融合与“逻辑”分离相结合架构的演进，实现以大统一融合架构形态实现对典型企业IT应用整合及性价比最优化支撑（见图6-7）。

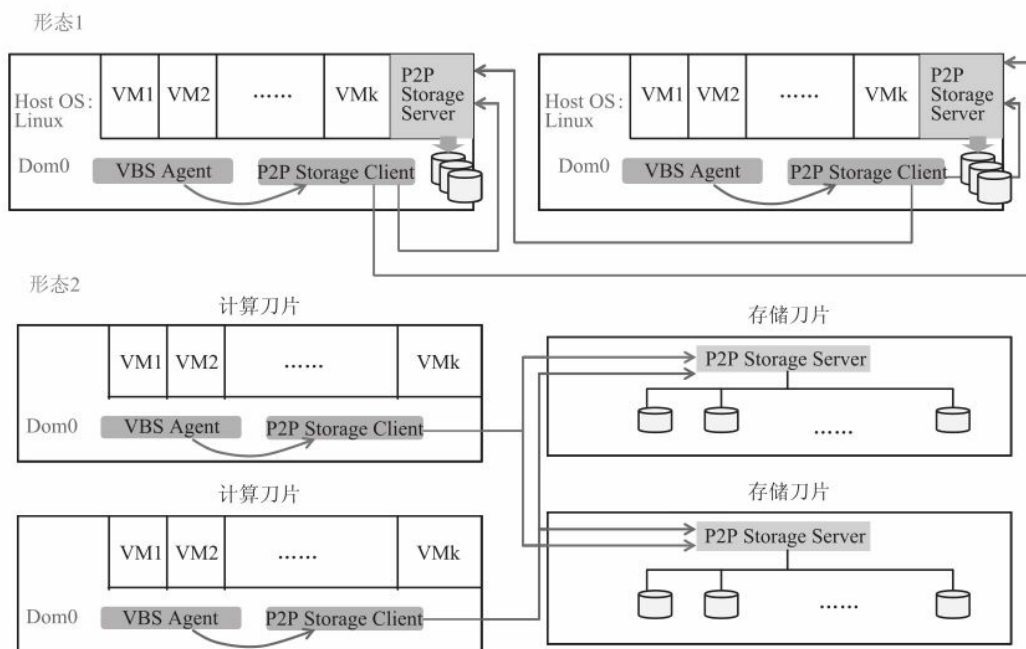


图6-7 Scale-out架构

在这种Scale-out架构与计算和网络融合后，便形成了一种更加高效的一体化分布式存储与分布式计算架构（见图6-8）。

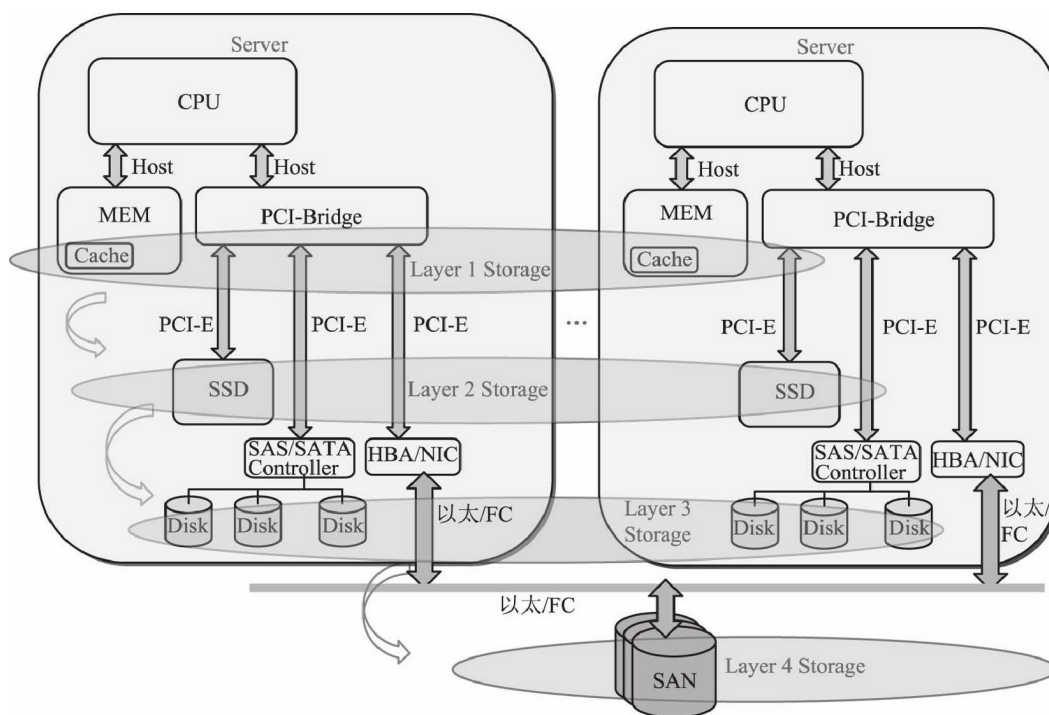


图6-8 融合一体化架构

一体化分布式存储架构具有以下鲜明特点。

➤ 一体化系统内，设置Layer1~Layer3共三层信息存储，基于分布式存储软件引擎完全水平拉通，且支持基于强一致的跨服务器数据可靠型。

- Layer1 Storage（内存）：时延100 ns（本地）~100 us（远地）
- Layer2 Storage（SSD）：时延10 us（本地）~300 us（远地）
- Layer3 Storage（DAS）：时延5ms（本地）~10 ms（远地）
- Layer4 Storage（SAN）：时延5ms（本地）~10 ms（远地）

➤ 通过上述各层Storage的热点数据读写推至更上一层Storage，实现数据I/O吞吐及整个系统性能的大幅提升。

➤ Layer1 Storage（内存）尽管吞吐率及时延优势明显，但容量和功耗成本过高，因此对于多数应用只能作为Cache，必须基于Cache算法（FIFO、LRU、LFU等）管理内存与下一级存储（SSD/DAS/SAN）之间的读写数据刷新。

➤ Layer2 Storage（SSD）既可以作为Cache（针对数据量超出分布式Cache容量的场景），也可以作为最终存储（纯SSD，针对数据容量不大的场景），为提升容量效率，未来需考虑进一步引入硬加速的块级去重引擎。

➤ Layer3 Storage（DAS）相比Layer 4 Storage，时延基本相同，因此作为Cache，意义不大，但可以作为Layer 4 Storage（SAN）的替代或补充（部分数据放置在外置SAN，部分数据放置在Layer 3 Storage），从而达到降低用户购置及维护外置存储的TCO。

➤ Layer 2 Storage（SSD）作为最终存储介质，因SSD不存在机械损坏故障风险，因此关键在于如何通过优化的Cache算法，将随机写I/O串行化，从而有效降低SSD写放大率，提升SSD寿命，用于中高端数据库等业务。

➤ Layer 3 Storage（DAS）作为最终存储介质，与Layer 4 Storage一样，需面对硬盘机械振动、磨损、环境影响等特殊因素的制约，因此充分借鉴和共享Layer 4 Storage在硬盘故障检测与修复方面的

长期经验与成果积累。

➤ 针对新建私有云/公有云的场景推荐采用Layer 3 Storage；针对IT平台替换或改造项目，可考虑借助存储虚拟化充分重用现有的外置Layer 4 Storage，同时将新产生的业务数据部署在Layer 3 Storage上，也可在外置存储退网前将其数据向Layer 3逐步无缝平滑迁移。

➤ 基于内存/SSD网格的计算近端I/O加速：SSD应用加速需要靠近服务器和应用侧，这已经成为业界共识。业界最具代表性的SSD加速产品大多是单机版，不支持分布式缓存一致性，很多应用场景下无法使用，比如：无法支持共享磁盘环境的active/active集群，如双机、数据库集群；无法支持虚拟机集群的动态资源调度和虚拟机迁移功能。而分布式存储引擎利用有最先进的分布式集群技术，可以很好地解决传统架构中热点容量不足的问题。同时，其也可与IPSAN配合，形成高速缓存和外置低速的互补。

➤ 基于Scale-out计算、存储融合架构的I/O性能提升：针对随机IOPS读写，基于分布式存储软件，各服务器内存Cache总容量相比集中式SAN机头的Cache容量增加可达5倍以上，从而使热点数据访问命中率与读写效率提升3~5倍；分布式存储可以采用大容量低成本SATA硬盘提供与SAS/FC硬盘持平的性能，而且有效容量更大；在针对大文件对象的顺序读写方面，分布式存储为App实例或VM提供并发读写服务，使得突发MBPS提升3~5倍以上。

一体化分布式存储架构与Google那种“Data Center as a Computer”的区别是，后者是面向海量搜索业务的计算/存储垂直整合数据中心，前者是面向企业IT核心业务及电信业务的计算存储垂直整合的高性能、高可扩展的IT平台。

业界典型的分布式存储技术主要有分布式文件系统存储、分布式对象存储和分布式块设备存储等几种形式。分布式存储技术及其软件产品已经日趋成熟，并在IT行业得到了广泛的使用和验证，例如互联网搜索引擎中使用的分布式文件存储，商业化公有云中使用的分布式块存储等。分布式存储软件系统具有以下特点。

➤ 高性能：分布式哈希数据路由，数据分散存放，实现全局负载均衡，不存在集中的数据热点和大容量分布式缓存。

➤ 高可靠：采用集群管理方式，不存在单点故障，灵活配置多数数据副本，不同数据副本存放在不同的机架、服务器和硬盘上，单个物理设备故障不影响业务的使用，系统检测到设备故障后可以自动重建数据副本。

➤ 高扩展：没有集中式机头，支持平滑扩容，容量几乎不受限制。

➤ 易管理：存储软件直接部署在服务器上，没有单独的存储专用硬件设备，通过Web UI的方式进行软件管理，配置简单。

6.4.2 分布式存储系统的架构

1. 分布式存储池的概念

分布式存储系统把所有服务器的本地硬盘组织成若干个资源池，基于资源池提供创建/删除应用卷（Volume）、创建/删除快照等接口，为上层软件提供卷设备功能。

分布式存储系统资源池具有如下特点（见图6-9）：

➤ 每块硬盘分为若干个数据分片（Partition），每个Partition只属于一个资源池，Partition是数据多副本的基本单位，也就是说多个数据副本指的是多个Partition。

➤ 系统自动保证多个数据副本尽可能分布在不同的服务器上（服务器数大于数据副本数时）。

➤ 系统自动保证多个数据副本之间的数据强一致性。

➤ Partition中的数据以Key-Value的方式存储。

➤ 对上层应用提供卷设备（Volume），没有LUN的概念，使用简单。

➤ 系统自动保证每个硬盘上的主用Partition和备用Partition数量是相当的，避免出现集中的热点。

➤ 所有硬盘都可以用做资源池的热备盘，单个资源池最大支持数百上千块硬盘。

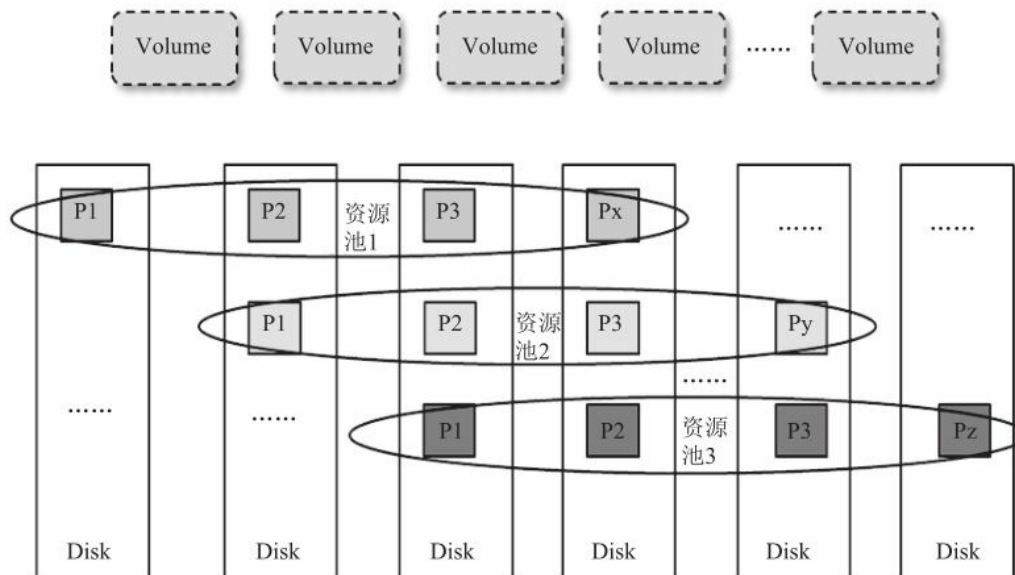


图6-9 分布式存储系统存储资源池

2. 分布式存储系统的功能框架

分布式存储系统采用分布式集群控制技术和分布式Hash数据路由技术，提供分布式存储功能特性。分布式存储系统功能架构图如图6-10所示。



图6-10 分布式存储系统功能架构

分布式存储系统功能模块具体如下。

- 存储接口层：通过SCSI驱动接口向操作系统、数据库提供卷设备。
- 存储服务层：提供各种存储高级特性，如快照、链接克隆、精简配置、分布式Cache、容灾备份等。
- 存储引擎层：分布式存储系统存储基本功能，包括管理状态控制、分布式数据路由、强一致性复制技术、集群故障自愈与并行数据重建子系统等。
- 存储管理层：实现分布式存储系统软件的安装部署、自动化配置、在线升级、告警、监控和日志等OM功能，同时对用户提供Portal界面。

3. 分布式存储系统的应用场景

分布式存储系统尤其适合计算和存储融合一体化系统。传统的虚拟化方式是在相互分离的计算、存储和网络设备上叠加了一层虚拟化软件。这

种方式虽然可以提升资源利用率，但是由于系统的复杂性，并不能简化各类基础设施的运维成本。融合一体化系统真正实现了计算、存储和网络设备的深度融合，硬件设备与虚拟化软件平台的一体化。一体化IT系统采用分布式存储系统把计算服务器的本地硬盘组织成一个类似SAN设备的虚拟存储池，对上层应用提供存储功能。

在IT平台中，分布式存储系统替代了传统的外置存储设备。适合使用分布式存储系统的应用场景。

- VDI、OA应用。其典型特点是：容量共享精简分配，性能共享分时复用，计算和存储配比相对均衡，成本性价比要求高。
- 虚拟化环境混合应用。其典型特点是：容量共享需求明显，多应用混合负载，线性扩展。
- OLAP应用。其典型特点是：大并发吞吐量，计算和存储带宽要求高。
- OLTP应用。其典型特点是：IOPS并发度高。

6.4.3 分布式存储关键技术：性能提升技术

性能卓越

分布式存储系统通过创新的架构把分散的、低速的SATA/SAS机械硬盘组织成一个高效的类SAN存储池设备，提供比SAN设备更高的I/O，把性能发挥到了极致。

分布式存储系统一般支持使用SSD替代HDD作为高速存储设备，支持使用InfiniBand网络替代GE/10GE网络提供更高的带宽，为对性能要求极高的大数据量实时处理场景提供完美的支持。

分布式存储系统采用无状态的分布式软件机头，机头部署在各个服务器上，无集中式机头的性能瓶颈。单个服务器上软件机头只占用较少的CPU资源，却能提供比集中式机头更高的IOPS。其实现了计算和存储的融合，缓存和带宽都均匀分布到各个服务器节点上。

分布式存储系统集群内各服务器节点的硬盘使用独立的I/O带宽，不存在独立存储系统中大量磁盘共享计算设备和存储设备之间有限带宽的问题。其将服务器部分内存用做读缓存，NVDIMM用做写缓存，数据缓存均匀分布到各个节点上，所有服务器的缓存总容量远大于采用外置独立存储的方案。即使采用大容量低成本的SATA硬盘，分布式存储系统仍然可以发挥很高的I/O性能，整体性能提升1~3倍，同时提供更大的有效容量。

全局负载均衡

分布式存储系统的实现机制保证了上层应用对数据的I/O操作均匀分布在不同服务器的不同硬盘上，不会出现局部的热点，实现全局负载均衡。

第一，系统自动将数据块打散存储在不同服务器的不同硬盘上，冷热不均的数据会均匀分布在不同的服务器上，不会出现集中的热点。

第二，数据分片分配算法保证了主用副本和备用副本在不同服务器和不同硬盘上的均匀分布，换句话说，每块硬盘上的主用副本和备副本数量是均匀的。

第三，扩容节点或者故障减容节点时，数据恢复重建算法保证了重建后系统中各节点负载的均衡性。

分布式SSD存储

分布式存储系统通过支持高性能应用设计的SSD存储系统，可以拥有比传统的机械硬盘（SATA/SAS）更高的读写性能。特别是PCIe卡形式的SSD，会带来更高的带宽和I/O，采用PCIe 2.0 x8的接口，可以提供高达3.0GB的读/写带宽。SSD I/O性能可以达到4KB数据块，100%随机，提供高达600K的持续随机读IOPS和220K的持续随机写IOPS。

SSD存在一个普遍的问题，就是写寿命问题，在采用SSD的时候，分布式SSD存储系统通过以下措施增强了可靠性（见图6-11）。

➤ 内嵌的ECC检错/纠错引擎和RAID5引擎，数据通道间形成二维的检错/纠错机制；

- 内置DATA Scrubbing引擎定时检测存储数据，提前预防数据错误的产生；
- 通道间使用Dynamic RAID算法，实现通道间的资源共享，确保在芯片坏块过多甚至是多个芯片故障的情况下均能正常工作；
- 内部实现冷热数据分类与管理，配合先进的磨损算法，最大程度地提升回收效率，降低写磨损，从而提升SSD的使用寿命。

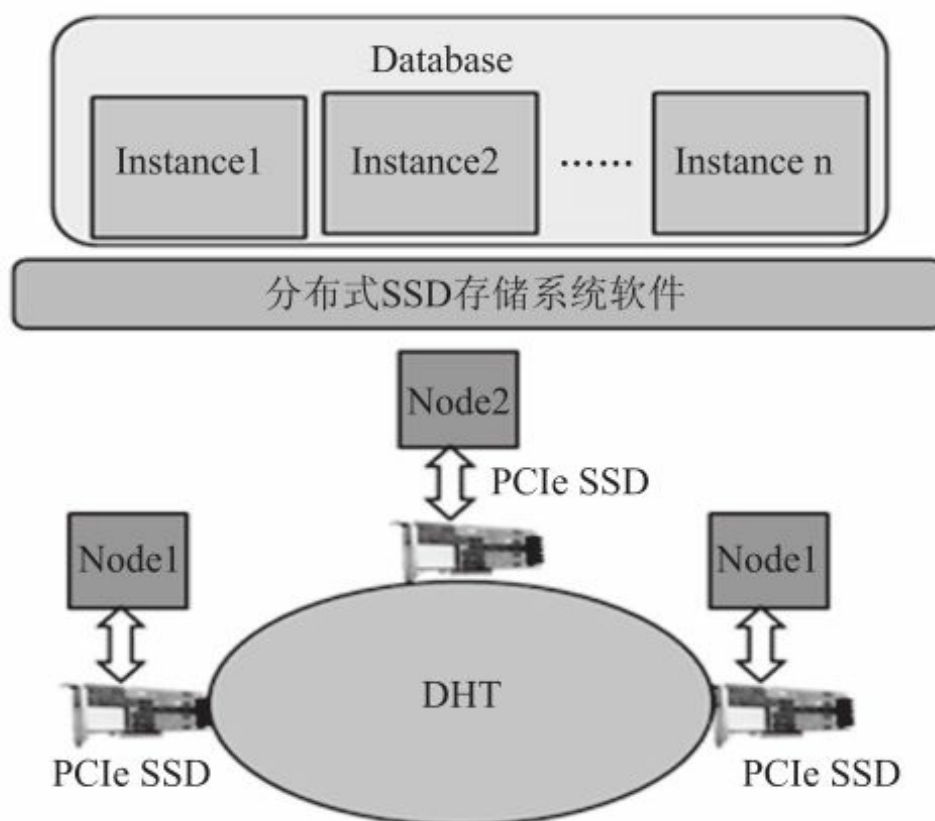


图6-11 分布式存储系统支持分布式SSD存储系统

高性能快照

分布式存储系统提供了快照机制，将用户的逻辑卷数据在某个时间点的状态保存下来，后续可以作为导出数据、恢复数据之用。

分布式存储系统快照数据基于DHT机制，快照不会引起原卷性能下降。针对一块容量为2TB的硬盘，完全在内存中构建索引需要几十MB空

间，通过一次Hash查找即可判断有没有做过快照，以及最新快照的存储位置，因此效率很高（见图6-12）。

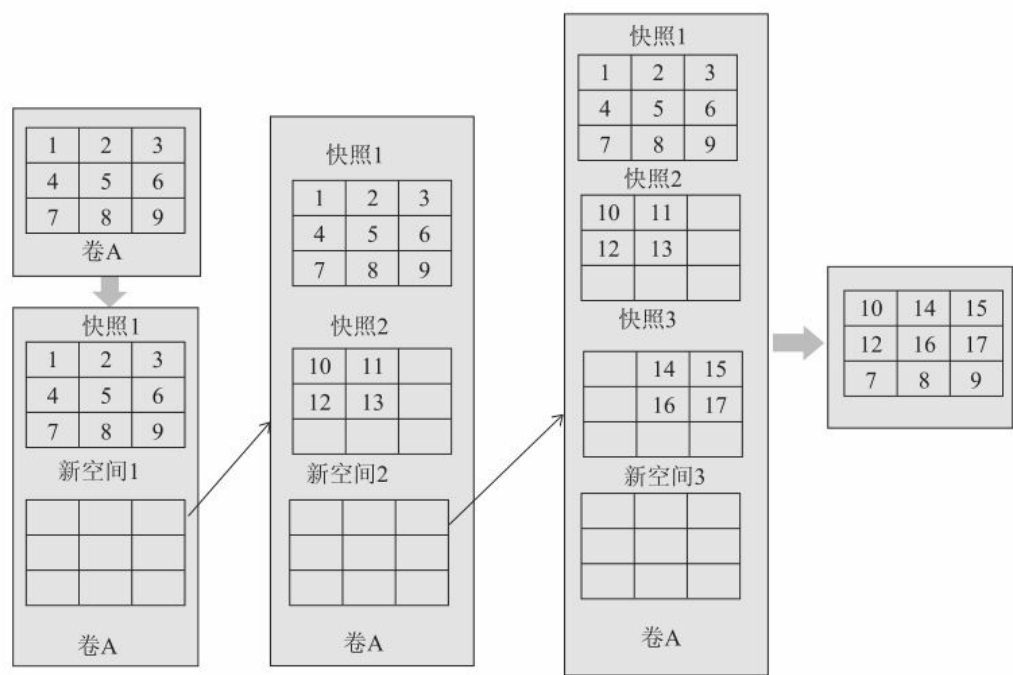


图6-12 分布式存储系统快照

高性能链接克隆

分布式存储系统可以基于增量快照提供链接克隆机制，基于一个快照创建出多个克隆卷，各个克隆卷刚创建出来时的数据内容与快照中的数据内容一致，后续对于克隆卷的修改不会影响原始的快照和其他克隆卷。分布式存储系统通过支持批量进行虚拟机卷部署，可以在秒级批量创建上百个虚拟机卷。克隆卷可支持创建快照、从快照恢复以及再次作为母卷进行克隆操作（见图6-13）。

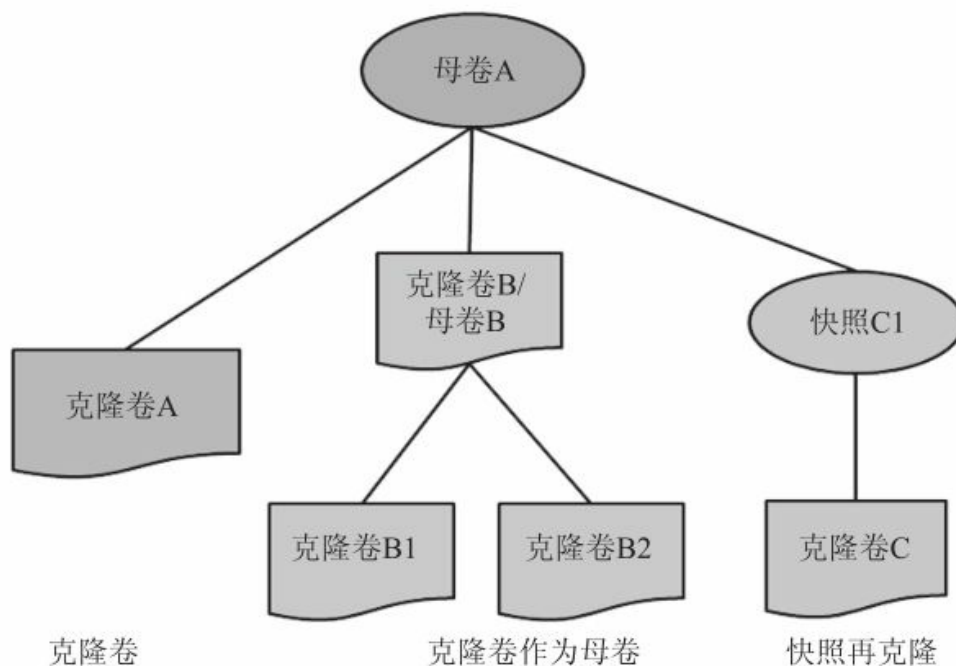


图6-13 分布式存储系统链接克隆

高速InfiniBand网络

为消除分布式存储环境中存储交换瓶颈，分布式存储系统可以部署为高带宽应用设计的InfiniBand网络。InfiniBand网络可以为分布式存储系统带来以下特性：

- 56Gbps FDR InfiniBand，超高速互联；
- 标准成熟多级胖树组网，平滑容量扩容；
- 近似无阻塞通信网络，数据交换无瓶颈；
- 纳秒级通信时延，计算存储信息及时传递；
- 无损网络QoS，数据传送无丢失；
- 主备端口多平面通信，冗余通信无忧；
- 单口56Gbps带宽，完美配合极速SSD存储吞吐，性能无限。

6.4.4 分布式存储关键技术：简化管理技术

分布式存储系统采用的分布式集群架构，天然支持无性能损耗的弹性扩展。

DHT数据路由：分布式存储系统采用DHT（Distribute Hash Table，分布式哈希表）路由数据算法。每个存储节点负责存储一小部分数据，基于DHT实现整个系统的寻址和存储。

DHT算法具有以下特点。

- **均衡性（Balance）：**数据能尽可能地分布到所有的节点中，这样可以使得所有节点负载均衡。
- **单调性（Monotonicity）：**当有新节点加入到系统中时，系统重新做数据分配，原来的数据存储位置不需要很大的调整。

由于分布式存储系统存储路由采用分布式哈希算法，使得存储系统具有如下特点（见图6-14）。

- **快速达到负载均衡：**新加入节点只需要搬移很少部分数据分片即可达到负载均衡。
- **数据高可靠：**灵活配置的分区分配算法，避免多个数据副本位于同一个服务器、同一个磁盘上。

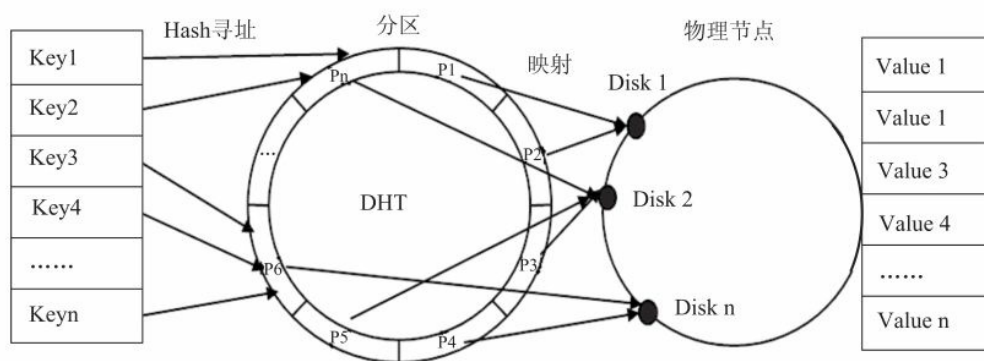


图6-14 分布式存储系统DHT数据路由

平滑扩容节点

分布式存储系统的分布式架构具有良好的可扩展性，支持超大容量的存储（见图6-15）。

- DHT算法保证了扩容后不需要做大量的数据搬迁，可以快速达到负载均衡状态。
- 扩展计算节点可以同步扩容存储空间，新扩展节点和原有节点可构成统一的资源池进行使用。
- 分布式存储系统分布式系统的带宽和Cache均匀分布在各个节点上，带宽和Cache不会随着节点的扩容而减少。

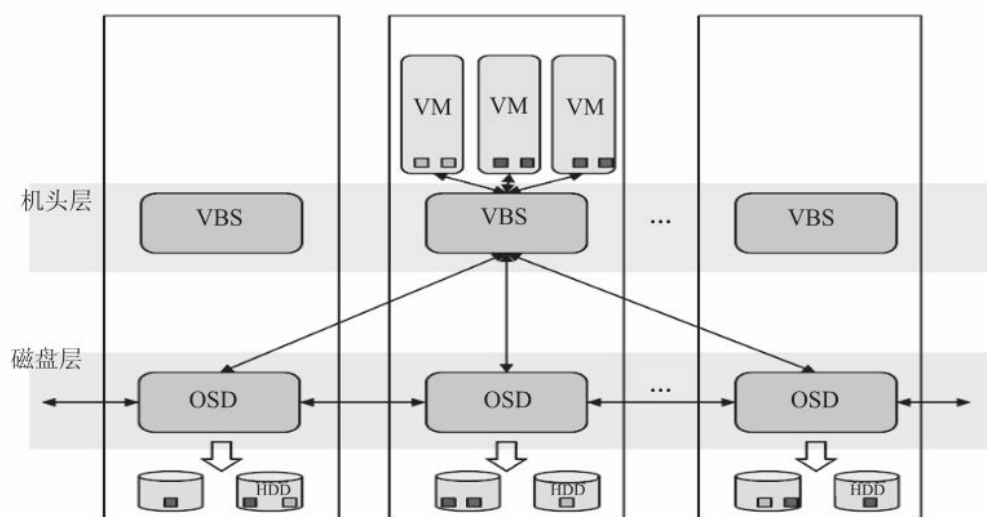


图6-15 分布式存储系统平滑扩容节点

资源按需使用

分布式存储系统提供了精简配置机制，为用户提供比实际物理存储更多的虚拟存储资源。相比直接分配物理存储资源，可以显著提高存储空间利用率。

采用分布式Hash技术，天然支持分布式自动精简配置（Thin Provisioning），无需预先分配空间。

精简配置（Thin Provisioning）无任何性能下降（IPSAN扩展空间时需要耗费额外的性能），如图6-16所示。

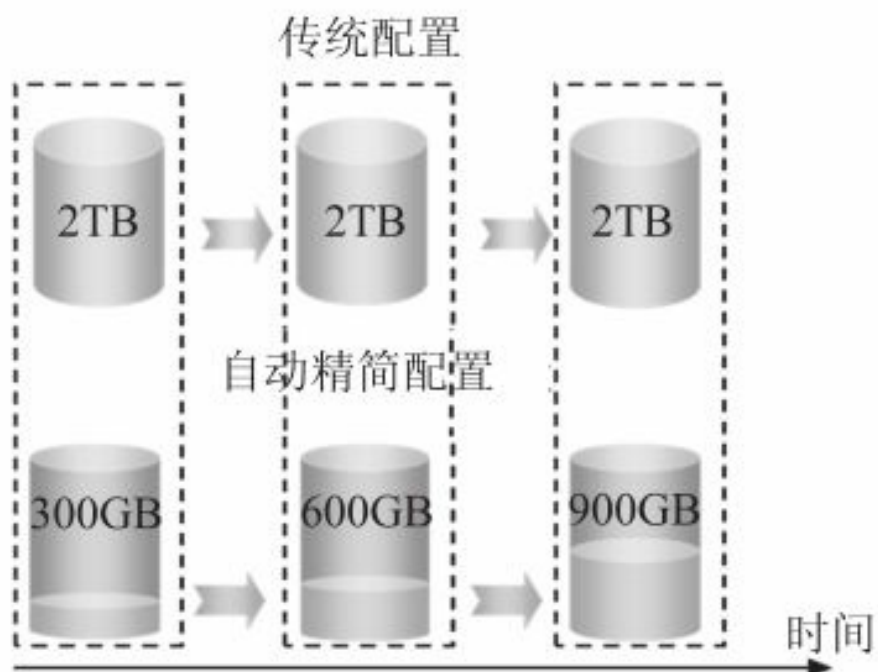


图6-16 分布式存储系统自动精简配置

统一的Web管理界面

分布式存储环境，在大规模场景下涉及设备众多，各组件的运行维护需要通过自动化形式完成，尽量减少人工干预。用户从Portal界面可以查看系统监控（KPI指标）、告警事件和存储池状态等，操作维护简单，有力帮助分布式系统在传统企业落地部署（见图6-17）。



图6-17 分布式存储系统Web UI界面样例

广泛兼容能力

其一般要求分布式存储系统采用通用x86服务器平台，在软件上支持通用的操作系统、数据库系统及虚拟化软件。

6.4.5 分布式存储关键技术：安全可靠增强技术

集群管理

分布式存储系统的分布式存储软件采用集群管理方式，规避单点故障，一个节点或者一块硬盘故障自动从集群内隔离出来，不影响整个系统业务的使用。

集群内选举进程Leader，Leader负责数据存储逻辑的处理，当Leader出现故障，系统自动选举其他进程成为新的Leader。

多数据副本

分布式存储系统存储系统中没有使用传统的RAID模式来保证数据的可靠性，而是采用了多副本备份机制，即同一份数据可以复制保存多个副本。在数据存储前，对数据进行分片，分片后的数据按照一定的规则保存集群节点上。

如图6-18所示，对于服务器Server1的磁盘Disk1上的数据块P1，它的数据备份为服务器Server2的磁盘Disk2上P1'，P1和P1'构成了同一个数据块的两个副本。

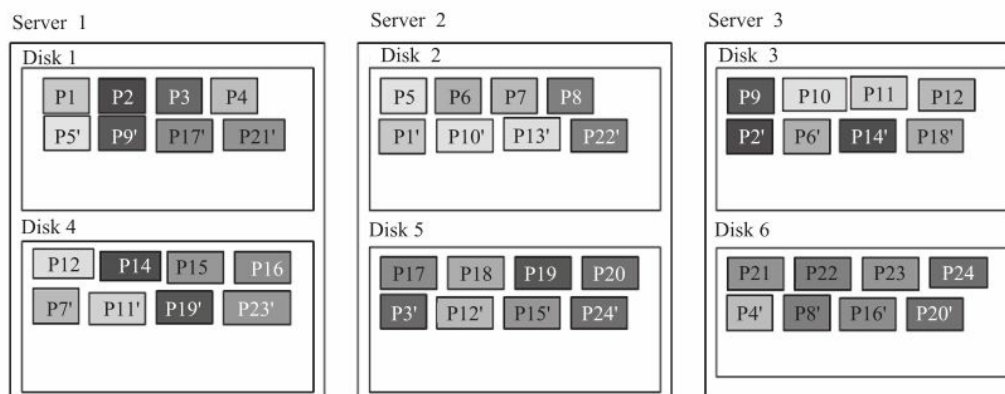


图6-18 分布式存储系统多数据副本

数据一致性

数据一致性的要求是：当应用程序成功写入一份数据时，后端的几个数据副本必然是一致的，当应用程序再次读取时，无论在哪个副本上读取，都是之前写入的数据，这种方式也是绝大部分应用程序所希望的。

保证多个数据副本之间的数据一致性是分布式存储系统的重要技术点，分布式存储系统采用强一致性复制技术确保各个数据副本的一致性，一个副本写入，多个副本读取。

分布式存储系统还支持Read Repair机制。Read Repair机制是指在读数据失败时，会判断错误类型，如果是磁盘扇区读取错误，可以通过从其他副本读取数据，然后重新写入该副本的方法进行恢复，从而保证数据副本总数不减少。

快速数据重建

分布式存储系统内部需要具备强大的数据保护机制。数据存储时被分片打散到多个节点上，这些分片数据支持分布在不同的存储节点、不同的机柜之间，同时数据存储时采用多副本技术，数据会自动保存多份，每一个分片的不同副本也被分散保存到不同的存储节点上。在硬件发生故障导致数据不一致时，分布式存储系统通过内部的自检机制，通过比较

不同节点上的副本分片，自动发现数据故障。发现故障后启动数据修复机制，在后台修复数据。由于数据被分散到多个不同的存储节点上保存，数据修复时，在不同的节点上同时启动修复，每个节点上只需修复一小部分数据，多个节点并行工作，有效避免单个节点修复大量数据所产生的性能瓶颈，对上层业务的影响做到最小化。数据故障自动恢复流程如图6-19所示。

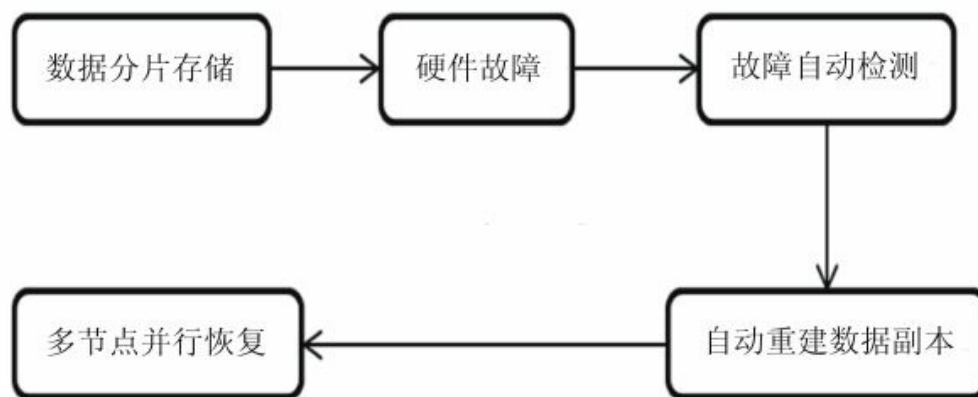


图6-19 分布式存储系统数据重建流程

分布式存储系统需要支持并行、快速故障处理和重建：

- 数据分片在资源池内被打散，硬盘出现故障后，可在资源池范围内自动并行重建；
- 数据分布上支持跨服务器或跨机柜，不会因某个服务器故障而导致数据不可访问；
- 扩容时可以自动进行负载均衡，应用无需调整即可获得更大的容量和性能。

掉电保护

分布式存储系统运行过程中可能会出现服务器突然下电的情况，分布式存储系统在内存中的元数据和写缓存数据会随着掉电而丢失，需要使用NVDIMM非易失内存来保存和恢复元数据和缓存数据。

部署分布式存储系统软件的每一台服务器要求配备NVDIMM内存条，服务器掉电时会把元数据和缓存数据写入NVDIMM的Flash中，上电后

又会把Flash中的数据还原到内存中。

分布式存储系统能够识别出系统中的NVDIMM内存，并把需要保护的数据按照内部规则存放在NVDIMM中，以便提供掉电保护功能。

第7章 云接入的关键技术架构与应用

我们知道，一个发电厂建设好后，若每家每户使用电的话，还需要架设长途电缆、建设变电站、在居民楼设置变压器、在每家每户铺设电线、安置插座和开关，设置电表，这样才能够将电力传送到千家万户。同样地，第4章、第5章和第6章分别讲述了计算虚拟化、网络虚拟化和存储虚拟化，利用这些技术，我们已经将分布在不同地方的数据中心进行虚拟化，构筑一个可以提供云服务的、很大的IT资源池，就是说，“IT电厂”已经建成。但是，对于最终的用户，要获得并使用IT资源，就要解决如何高效访问云计算系统以获取所需的IT资源，并获得满意的访问体验的问题，这就是云接入。

7.1 云接入的概述

7.1.1 什么是云接入

云接入是指从单一平台实现桌面和应用虚拟化，提供固定和移动终端融合接入的统一工作空间，帮助客户对固定办公和移动办公环境下的桌面、应用和数据进行统一管理、发布和聚合。它是一种基于云计算的终端用户计算模式。在这种模式中，所有的应用程序都在云数据中心运行，应用程序无需在终端上安装。用户通过终端云接入协议连接到云数据中心，并运行在云数据中心的程序，以获取程序运行结果。

随着企业信息化进程的不断深入，企业中增加了各种各样的电子设备。由于传统IT的束缚，企业IT团队依然要维护大量的传统PC，这不仅需要大量的人力物力，而且在进行外网接入以及异地登录的时候无法很好地保障企业数据安全。全球可连接互联网设备的出货情况显示，PC所占份额越来越小。2013年，智能手机出货量接近10亿部。据调查，美国人每天在智能手机上花1个小时，每天在平板电脑上花半个小时。移动带来了娱乐、通信、媒体和商务新方式。企业IT基础架构要不断适应这种新变化。

云接入很好地解决了这些问题，不仅可以快速地搭建企业IT基础架构，还可以快速对员工账户进行管理，实现跨平台作业。

桌面云是对云接入桌面这一侧重点的专门阐述。

7.1.2 云接入的作用和意义

云接入的业务价值很多，除了上面所提到的随时随地访问桌面以外，还有下面一些重要的业务价值。

数据上移，信息安全

传统桌面环境下，由于用户数据都保存在本地PC，因此内部泄密途径众多，且容易受到各种网络攻击，从而导致数据丢失。云接入桌面环境下，终端与数据分离，本地终端只显示设备，无本地存储，所有的桌面数据都集中存储在企业数据中心，无需担心企业的智力资产泄露。除此之外，TC的认证接入、加密传输等安全机制，保证了云接入桌面系统的安全可靠。

高效维护，自动管控

传统桌面系统故障率高，据统计，平均每400台PC机就需要一名专职IT人员进行管理维护，且每台PC维护流程（故障申报->安排人员维护->故障定位->进行维护）需要2~4个小时。

在云接入桌面环境下，可实现资源自动管控，维护方便简单，节省IT投资。

➤ 维护效率提升：云接入桌面云不需要前端维护，强大的一键式维护工具让自助维护更加方便，提高企业运营效率。使用云接入桌面后，每位IT人员可管理超过2000台虚拟桌面，维护效率提高4倍以上。

➤ 资源自动管控：白天可自动监控资源负载情况，保证物理服务器负载均衡；夜间可根据虚拟机资源占用情况，关闭不使用的物理服务器，节能降耗。

应用上移，业务可靠

在传统桌面环境下，所有的业务和应用都在本地PC上进行处理，稳定性仅99.5%，年宕机时间约21个小时。在云接入桌面方案中，所有的业

务和应用都在数据中心进行处理，强大的机房保障系统能确保全局业务年度平均可用度达99.9%，充分保障业务的连续性。各类应用的稳定运行，有效降低了办公环境的管理维护成本。

无缝切换，移动办公

在传统桌面环境下，用户只能通过单一的专用设备访问其个性化桌面，这极大地限制了用户办公地的灵活性。采用云接入桌面，由于数据和桌面都集中运行和保存在数据中心，用户可以不中断应用运行，实现无缝切换办公地点。

降温去噪，绿色办公

节能、无噪的TC部署，有效地解决了密集办公环境的温度和噪音问题。TC让办公室噪音从50分贝降低到10分贝，办公环境变得更加安静。TC和液晶显示器的总功耗大约60W左右，终端低能耗可以有效地减少降温费用。

资源弹性，复用共享

➤ 资源弹性：在云接入桌面环境下，所有资源都集中在数据中心，可实现资源的集中管控，弹性调度。

➤ 资源利用率提高：资源的集中共享，提高了资源利用率。传统PC的CPU平均利用率为5%~20%，在云接入桌面环境下，云数据中心的CPU利用率可控制在60%左右，提升了整体资源利用率。

安装便捷，部署快速

云接入桌面解决方案具有安装便捷、部署快速的特点。到客户现场后，只需服务器上电，进行云接入桌面软件的向导式安装，接通网络并进行相关业务配置即可进行业务发放，大幅度提高了部署效率。

7.1.3 云接入的挑战和需求

云接入的挑战和需求，主要集中在如何应用虚拟化技术为终端用户提供资源访问的便利性、安全性以及用户体验上，通过分析这些典型技术的特点，可以发现他们仍然存在如下一些难以解决的问题。

外设兼容性

在云接入桌面虚拟化项目中对外设的支持是非常普遍的，绝大多数虚拟桌面基础架构项目中都会遇到用户对外设的需求，但有时也非常棘手。众所周知，外设的云终端上接入，在后端做桌面识别，这就涉及将具有电器特性的硬件设备通过网络传输到后端的桌面中，并且设备本身的驱动是在前端还是后端，都需要桌面云厂家考虑，加上国内外设的多样性和不标准性，要在桌面云中支持具有多样性复杂性的外设，需要厂家有独特的外设支持技术。

视频体验

云接入桌面的计算和存储全都在数据中心，终端只负责键盘鼠标的I/O和显示的输出，此时云桌面的传输协议就显得尤为重要。普通办公桌面的传输没有什么问题，但是实际上用户可能有各种各样的业务需求，例如视频，这类业务在终端桌面就可能出现画面不流畅，终端画面出现马赛克，更别提播放三维动画了，尤其随着桌面互联网的发展，很多桌面虚拟化方案需要基于互联网部署，而给予互联网的传输效果更是大打折扣。这要求桌面云厂家在桌面传输协议上有独特的通道设计，通过不同的通道来处理不同的桌面显示，并且在带宽上能优化处理。

3D应用

虚拟化桌面固然有其诱人之处，但是目前主流的桌面虚拟化技术在3D图形设计方面很难满足客户的需求，这也使得传统虚拟化方案在制造行业、数字内容创作等行业遇到了难以逾越的瓶颈，再好的解决方案如果不能满足用户的实际需求也是空谈。

网络负载压力

局域网一般不存在太大的问题，但是如果通过互联网就会出现很多技术难题，由于桌面虚拟化技术的实时性很强，如何降低这些传输压力，是很重要的一环；虽然千兆以太网对数据中心来说是一项标准，但还没有广泛部署到桌面，目前还达不到虚拟化桌面对高带宽的要求。而且如果用户使用的网络出现问题，桌面虚拟化发布的应用程序不能运行，则会直接影响应用程序的使用，其对用户的影响也是无法估计的。

安全、部署效率和用户体验是移动办公的主要挑战

移动办公的主要挑战如图7-1所示。



图7-1 移动办公的主要挑战

云接入的关键需求

随着企业的发展，分支机构、办事处、连锁店等企业扩大经营造成员工分布广，需要一种便捷、灵活和具有跨地域性的办公方案，使员工无论身在何处，都能实现员工与员工之间、企业与业务伙伴之间的相互交流和沟通。各级政府机构服务观念在不断提高，也希望通过移动化的方式提高办公效率，降低管理成本，提升服务质量。

市场人员遍布全国各地，没有固定的办公场所，他们每次访问内部系统的终端和网络的地方都不一样，没有局域网环境，他们无法使用CRM来更新客户信息，也无法利用OA系统来实现办公自动化。

公司领导经常出差办公，在火车站、飞机场等地方随时需要查看、调用、审批内部的资料文档，并知道业务进展及生产线的进度，需要随时随地都能访问内部办公系统及生产管理系统的解决方案。

政府工作人员驻点调查路况信息、各分局等的数据采集等，需要有一种方式可以将采集到的重要信息及时传达给内部系统。

突发和意外情况，能在事件发生的最短时间内上报、传达给企业内部的相关人员，相关人员和领导层能不受地点的限制，快速、及时地对突发

和意外情况做出指示和决定。

随时随地办公，通过公网访问企业内部核心信息资源，就面临着非法访问、信息窃取等外部的安全威胁，就必须有相应的信息安全策略，在严格防止企业信息资源被非法窃取的同时，对合法的访问要提供方便。

移动性对后PC时代的成功至关重要。在IT组织希望满足终端用户对使用各种设备在家、旅途中和办公室的一致体验要求的同时，IT基础设施必须确保业务计算环境安全、易于管理并具备持续合规性。

云接入解决方案既要能提高终端用户的自由，又不能削弱IT系统控制力，它必须以下面三条关键原则为基础。

- 简化：将终端用户资产（包括操作系统、应用和数据）从计算小环境转变为集中式IT托管服务。
- 管理：为IT创建一个中心点，用于跨公有云和私有云实现终端用户对IT服务的访问，并能够控制终端用户具有访问权限及相应的安全级别。
- 连接：改善终端用户与IT服务及其他终端用户的连接性，且终端用户能够为手头上的任务自由选择最合适的设备 and 应用。

7.2 云接入的架构

云接入架构如图7-2所示。

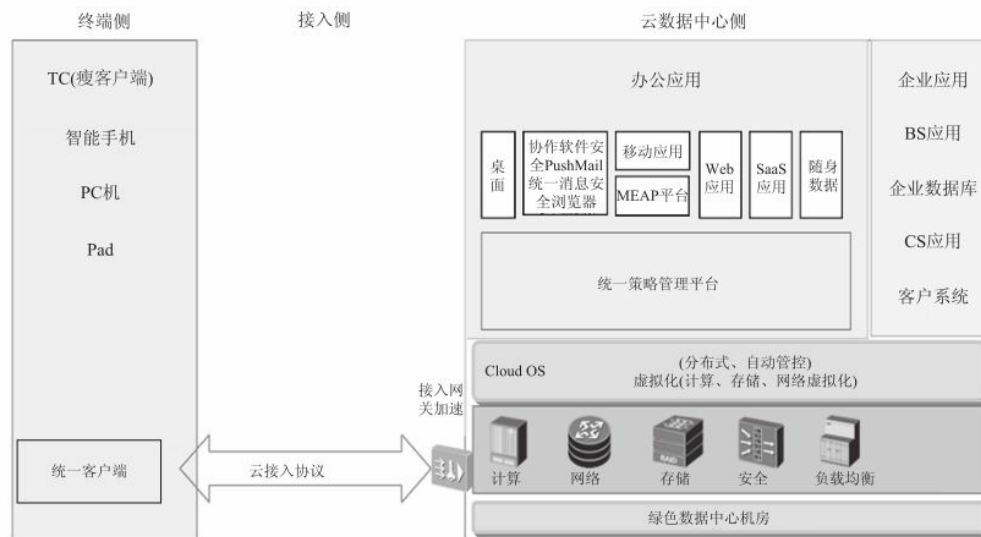


图7-2 云接入端对端架构图

- 终端侧：运行各种终端，在任何设备上随时随地访问用户应用、桌面、数据。
- 接入侧：云接入协议，实现从终端到云端的安全接入、加密传输、负载均衡、流量控制。
- 云数据中心侧：云接入统一策略管理，提供用户、应用、桌面、数据及策略管理及分发，并包含后台资源和软件的管理和配置。
- 云数据中心侧：云接入网关，提供云接入安全接入控制，协议加速。
- 办公应用：常用的办公应用如Windows办公桌面，统一通信协作软件，Web浏览器。
- 企业后台应用：支持企业日常业务运行的后台应用，如CRM/ERP/数据库。

7.3 云接入的典型应用

7.3.1 桌面云的概念和价值

云接入的典型应用就是我们最常见到的桌面云。

什么是桌面云，桌面云的定义是：“可以通过瘦客户端或者其他任何与网络相连的设备来访问跨平台的应用程序以及整个客户桌面。”也就是说我们只需要一个瘦客户端设备，或者其他任何可以连接网络的设备，通过专用程序或者浏览器，就可以访问驻留在服务器端的个人桌面以及各种应用，并且用户体验和我们使用传统的个人电脑是一模一样的。

桌面云的业务价值很多，除了上面提到的随时随地访问桌面以外，还有下面一些重要的业务价值。

集中化管理

在使用传统桌面的整体成本中，管理维护成本在其整个生命周期中占很大一部分，管理成本包括操作系统安装配置、升级、修复的成本，硬件安装配置、升级、维修的成本，数据恢复、备份的成本，以及各种应用程序安装配置、升级、维修的成本。在传统桌面应用中，这些工作基本上都需要在每个桌面上做一次，工作量非常大。对于那些需要频繁替换、更新桌面的行业来说，工作量就更大了。例如对于培训行业来说，他们经常需要配置不同的操作系统和运行程序来满足不同培训课程的需要，对于有上百台机器来说，这个工作量已经非常大了，而且这种工作还要经常变化内容。

在桌面云解决方案里，管理是集中化的，IT工程师通过控制中心管理成百上千的虚拟桌面，所有的更新、打补丁都只需要更新一个“基础镜像”。对于上面所提到的培训中心来说，管理维护就非常简单了：我们只需要根据课程的不同配置几个基础的镜像，然后不同的培训课程的学员就可以分别连接到这些不同的基础镜像，而且我们只需在这几个基础镜像上进行修改，只要重启虚拟桌面，学员就可以看到所有的更新，这样就大大节约了管理成本。

安全性提高

安全是IT工作中一个非常重要的方面，一方面各单位对自己有安全要求，另一方面政府对安全也有一些强制要求，一旦违反，后果非常严重。对于企业来说，数据、知识产权就是他们的生命，例如银行系统中

的客户的信用卡账号，保险系统中的用户详细信息，软件企业中的源代码等。如何保护这些机密数据不被外泄是许多公司IT部门经常面临的一个挑战。为此他们采用了各种安全措施来保证数据不被非法使用，例如禁止使用USB设备，禁止使用外面的电子邮件等。对于政府部门来说，数据安全也是非常重要的，英国不久前就发生了某政府官员的笔记本丢失，结果保密文件被记者得到，这个官员不得不引咎辞职。

在桌面云解决方案里，首先，所有的数据以及运算都在服务器端进行，客户端只显示其变化的影像，所以在不需要担心在客户端出现非法窃取资料行为，我们在电影里面看到的商业间谍拿着U盘疯狂地拷贝公司商业机密的情况再也不会出现了。其次，IT部门根据安全挑战制作出各种各样的新规则，这些新规则可以迅速地作用于每个桌面。

应用更环保

如何保护我们的有限资源，怎样才能消耗更少的能源，这是现在各国科学家在不断探索的问题。因为在我们地球上的资源是有限的，不加以保护的话很快会陷入无资源可用之困境。现在全世界都在想办法减少碳排放量，为之也采取了很多措施，例如利用风能等更清洁的能源等。但是传统个人计算机的耗电量是非常惊人的，一般来说，每台传统个人计算机的功耗在200W左右，即使处于空闲状态时，耗电量也至少在100W左右，按照每天10个小时，每年240天工作来计算，每台计算机桌面的耗电量在480度左右，一个具有1万桌面的中型企业，仅PC年耗电量就会达到480万度。除此之外，为了冷却这些计算机在使用中产生的热量，我们还必须使用一定的空调设备，这些能量的消耗也是非常大的。

采用桌面云解决方案以后，每个瘦客户端的电量消耗在16W左右，只有原来传统个人桌面的8%，所产生的热量也将大大减少。

总拥有成本减少

IT资产的成本包括很多方面，初期购买成本只是其中的一小部分，其他还包括整个生命周期里的管理、维护、能量消耗等方面的成本，硬件更新升级的成本。从上面的描述中我们可以看到相比传统个人桌面而言，桌面云在整个生命周期里的管理、维护、能量消耗等方面的成本大大降低了，那么硬件成本又是怎样的呢？桌面云在初期硬件上的投资是比较大的，因为我们要购买新的服务器来运行云服务，但是由于传统桌面的更新周期是3年，而服务器的更新周期是5年，所以硬件上的成本基本相

当。由于软成本的大大降低，而且软成本在TCO中占有非常大的比重，所以采用云桌面方案总体TCO大大减少了。根据Gartner公司的预计，云桌面的TCO相比传统桌面可以减少40%。

7.3.2 桌面云的逻辑架构

桌面云解决方案包括7个逻辑部分：云终端、接入控制、桌面会话管理、云资源管理及调度、虚拟化平台、运维管理系统和硬件，如图7-3所示。

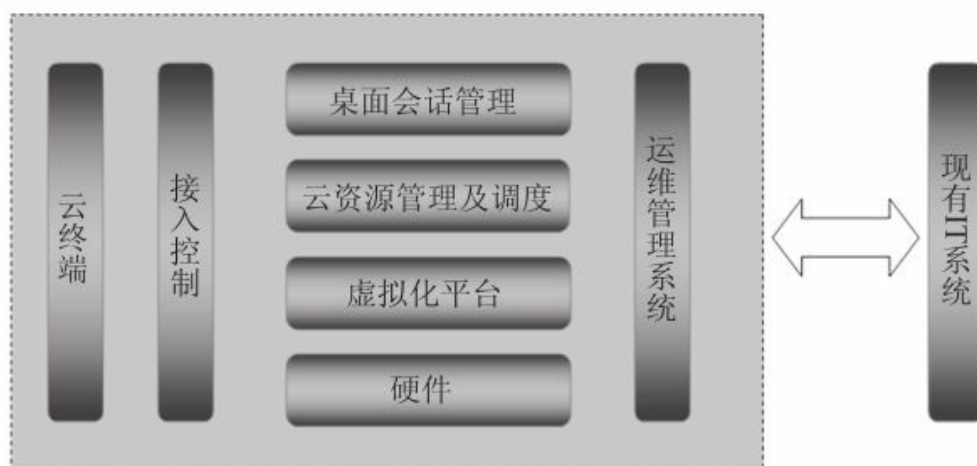


图7-3 桌面云解决方案逻辑图

云终端

云终端是在远端用于访问桌面云中虚拟桌面的特定的终端设备，包括瘦终端、软终端软件和各种手持终端等。

接入控制

接入控制用于对终端的接入访问进行有效控制，包括接入网关、防火墙、负载均衡器等设备。接入控制设备不是桌面云解决方案所必需的组成部分，可以根据客户的实际需求进行裁减。

桌面会话管理

桌面会话管理负责对虚拟桌面使用者的权限进行认证，保证虚拟桌面的使用安全，并对系统中所有虚拟桌面的会话进行管理。

云资源管理及调度

云资源管理是指根据虚拟桌面的要求，把桌面云中各种资源分配给申请资源的虚拟桌面，分配的资源包括计算资源、存储资源和网络资源等。

云资源调度是指根据桌面云系统的运行情况，把虚拟桌面从负载比较高的物理资源迁移到负载比较低的物理资源上，保证整个系统物理资源的均衡使用。

虚拟化平台

虚拟化平台是指根据虚拟桌面对资源的需求，把桌面云中各种物理资源虚拟化成多种虚拟资源的过程，这些虚拟资源可以供虚拟桌面使用，这些资源包括计算资源、存储资源和网络资源等。

硬件

硬件是指组成桌面云系统相关的硬件基础设施，包括服务器、存储设备、交换设备、机架、安全设备、防火墙、配电设备等。

运维管理系统

运维管理系统包括桌面云的业务运营管理和系统维护管理两部分，其中业务运营管理完成桌面云的开户、销户等业务发放过程，系统维护管理完成对桌面云系统各种资源的操作维护功能。

现有IT系统

现有IT系统指已经部署在现有网络中对桌面云有集成需求的企业IT系统，包括AD（Active Directory）、DHCP、DNS等。

7.3.3 桌面云典型应用场景

1. 办公桌面云解决方案

办公桌面云是指企业使用桌面云来进行正常的办公活动（如处理邮件、编辑文档等），同时提供多种安全方案，保证办公环境的信息安全。办公桌面云解决方案如图7-4所示。

特点

桌面云支持与企业已有的IT系统对接，充分利用已有的IT应用。比如利用已有的AD系统进行桌面云用户鉴权；在桌面云上使用已有的IT工作流；通过DHCP给虚拟桌面分配IP地址；通过企业的DNS来进行桌面云的域名解析等。

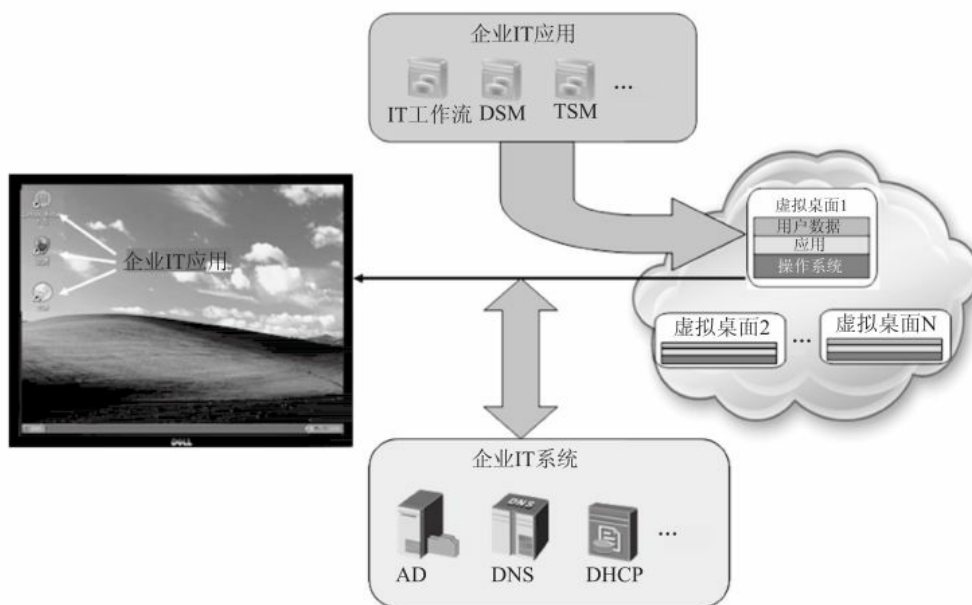


图7-4 办公桌面云解决方案

优势

- 减少投资，平滑过渡：充分利用已有的IT系统设备与IT应用，减少重复投资，做到平滑过渡。
- 可靠的信息安全机制：桌面云提供多种认证鉴权与管理机制，保证办公环境的信息安全。

2. 绿色座席解决方案

绿色座席解决方案如图7-5所示。

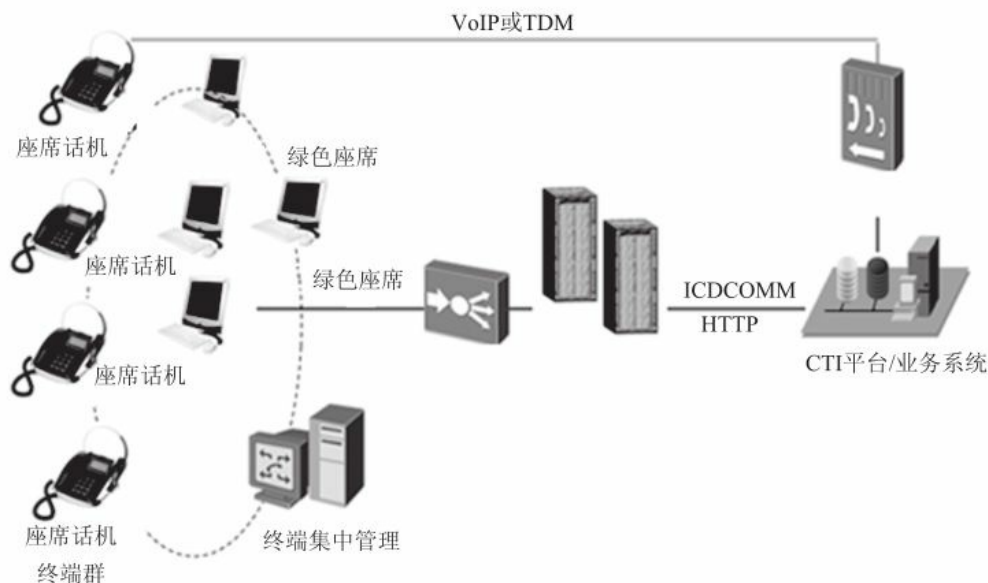


图7-5 绿色座席解决方案

特点

多数企业用户部署的呼叫中心越来越多地由TDM方式的语音解决方案演进到采用IP语音解决方案。

优势

- 支持平滑迁移：完善的呼叫中心平台和桌面云的集成方案，平滑迁移客户原有呼叫中心。
- 快速应用，优质语音：提供桌面应用的快速响应特点和优质的语音体验。
- 成本优化：同类应用的共享部署模式，大大节省了虚拟桌面实例的资源占用，方便维护、升级。采用TC终端替代传统PC，降低呼叫中心的噪音、电力消耗，为客户打造绿色呼叫中心。

3. 营业厅解决方案

企业营业厅系统划分为服务人员使用的桌面系统、业务办理的客户使用的自助系统。营业厅解决方案如图7-6所示。

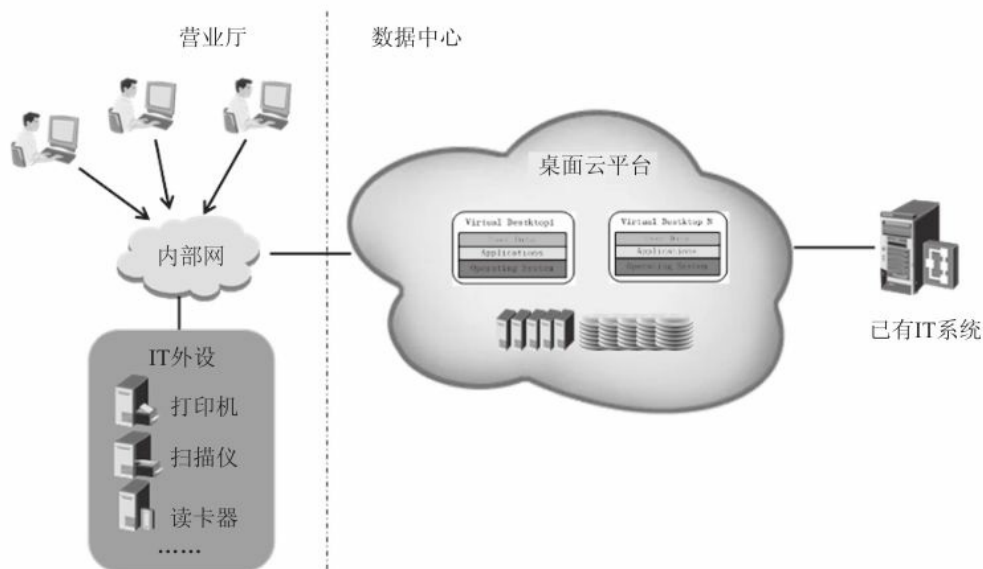


图7-6 营业厅解决方案

特点

营业厅解决方案针对营业厅分布地域广泛、网络连接质量差异大的特点，改进了桌面云系统的调度和连接优化配置，可以有效地克服网络闪断、网络速率抖动等恶劣条件，保证企业营业厅系统的高质量服务。

营业厅解决方案提供即插即用的外设终端接入方案，并通过预置的具备广泛兼容性的驱动插件支持常见的串口、并口、USB口外设，极大地降低了企业客户部署的难度。

根据企业营业厅的业务特点，其支持多种桌面系统认证方式。对于客户自助系统的桌面，其还可以支持免认证登录桌面系统、即时打印服务清单等功能。

优势

营业厅解决方案是针对各种营业厅推出的解决方案，营业厅解决方案具有如下优点。

- **利旧原有IT外设：**无需采购新的IT外设，兼容常见接口外设，并可对于外设驱动统一部署和管理，保证即插即用的客户体验。
- **快速软件安装部署：**运营软件通过云平台集中推送，做到大规

模快速软件安装部署，便于企业统一新业务上线。

➤ 支持客户自助系统：支持客户自助系统在桌面云的部署，可免认证使用企业为客户提供的系统，可即时打印服务清单等。

4. 网管维护解决方案

网管维护解决方案如图7-7所示。

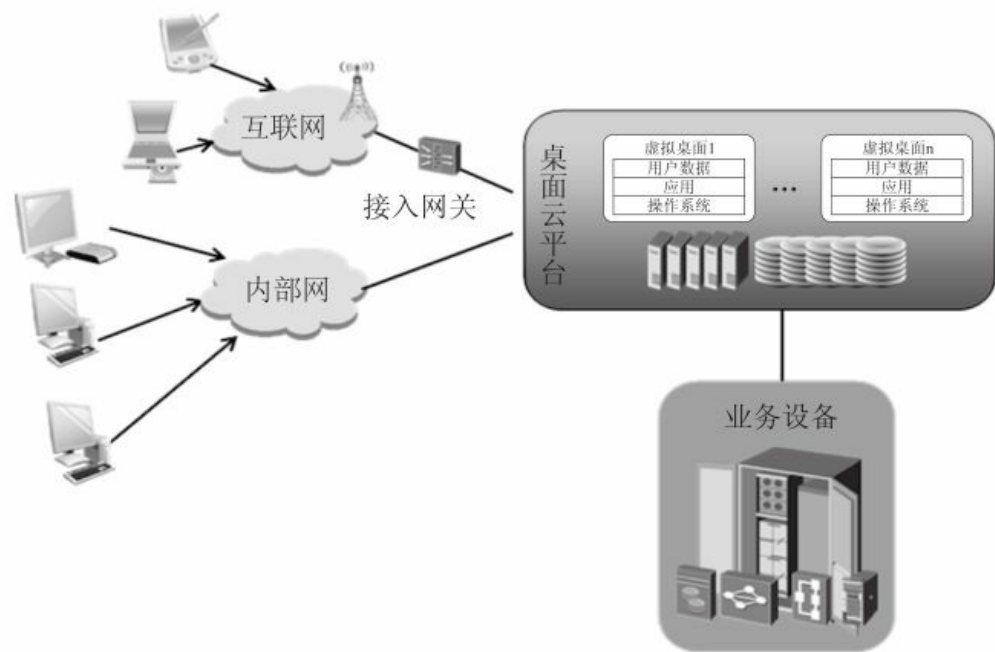


图7-7 网管维护解决方案

特点

桌面云网管维护解决方案针对网络管理的特点，定制了多种接入终端的接入程序，方便随时随地地接入进行网络状态分析与网络故障定位，对于重大问题，充分发挥企业网管专家的经验优势。

桌面云网管维护解决方案集成多种网管适配解决方案，无需既有网管系统进行改造，即可实现统一管理。

优势

网管维护解决方案是针对各类网管推出的解决方案，网管维护解决具有如下优点。

- 无缝接入：支持各种接入终端，包括多种手持终端（Android类、Windows Mobile类，iPhone OS类，iPad OS类，Embedded Linux类终端），可以实现无缝地、随时随地地接入以及远程维护和监控，有利于企业发挥维护专家的优势。
- 广泛支持多种类型的网管系统：支持远程维护非C/S、B/S架构的网管系统，使用近端观测程序，极大地减少现场维护需求。
- 整合零散的网管系统：企业现有网管系统无需改造，通过桌面云系统即可以实现网络的全集中管理，提高网管维护效率。

7.3.4 移动办公的概念和价值

移动办公是云接入的另一种常见的典型应用。

移动办公与移动办公系统

移动办公也称移动OA，是指利用手机、PDA或笔记本电脑等移动终端通过无线网实现移动办公的移动信息化手段，移动办公软件是实现移动办公的软件应用系统，它使得移动办公人员摆脱时间和场所局限，随时进行随身化的信息管理和沟通，从而有效提高管理效率，推动政府和企业效益增长。

移动办公是当今高速发展的通信业与IT业交融的产物，它将通信业在沟通上的便捷、在用户上的规模，与IT业在软件应用上的成熟、在业务内容上的丰富，完美结合到了一起，使之成为继电脑无纸化办公、互联网远程化办公之后的新一代办公模式。这种最新潮的办公模式，通过在手机、Pad上安装企业信息化软件，使得手机、Pad也具备了和电脑一样的办公功能，而且它还摆脱了必须在固定场所固定设备上办公的限制，对企业管理者和商务人士提供了极大的便利，为企业和政府的信息建设提供了全新的思路 and 方向。它不仅使得办公变得随心、轻松，而且借助手机通信的便利性，使得使用者无论身处何种紧急情况下，都能高效、迅捷地开展工作，对于突发性事件的处理、应急性事件的部署有极为重要的意义。

移动办公系统，是一套建立以手机等便携终端为载体实现的移动信息化系统，系统将智能手机、无线网络、OA系统三者有机结合，实现与任

何办公地点和办公时间的无缝接入，提高了办公效率。它可以连接客户原有的各种IT系统，包括OA、邮件、ERP以及其他各类个性业务系统，可通过手机操作、浏览、管理公司的全部工作事务，同时提供一些无线环境下的新功能。其设计目标是帮助用户摆脱时间和空间的限制，随时、随地、随意地处理工作，提高效率、增强协作。

移动办公的意义

移动办公是一种新型的低碳办公模式，能为企业和社会节约资源，减少废气排放。

移动办公是一种无纸化低碳办公模式。移动办公系统通常能支持pdf、jpg、doc、xls等文件格式，这些文件格式基本覆盖了大多数企业内的文件审批格式。现在，领导使用手机等移动终端即可打开各种待审核和待审批公文，远程进行批复。业务人员在与客户会谈前总是需要先打印出各种准备资料，而在使用装有移动办公系统的手机端之后，只需在会谈期间根据需要随时进入公司系统进行查阅，同时运用PPMEET等相关视频会议软件进行会议的召开。

移动办公是一种电能消耗极少的低碳办公模式。与大约每小时消耗0.2~0.3度电的PC相比，手机消耗的电量几乎可以忽略不计。企业是电能消耗大户，对于移动办公任务较多的企业来说，通过移动办公每月可节约的电量将是一个不小的数字，同时可为社会节约电力资源。

移动办公大大减少了用户在交通工具所需的汽柴油等燃料方面的消耗，同时减少含有大量二氧化碳气体的尾气排放。这是一种典型的污染物排放少的低碳办公方式。业务型员工和领导移动办公任务重，在外忙碌了一天之后往往还需要返回公司，处理一些收尾的工作，例如将业务信息录入系统，查询最新的通知公告等。现在，移动办公系统省去了用户返回公司的必要，大大减少了乘坐交通工具所需的燃料方面的消耗。

如今，移动办公已经不仅是一种节约能源、减少二氧化碳污染的低碳办公模式，还是提高员工办公效率、提高企业综合竞争力、提升公众形象的一种手段。

7.3.5 移动办公的逻辑架构

主流解决方案：原生应用与远程应用

图7-8 对比了两种主流的移动办公解决方案，我们可以从中看出，远程应用在安全、维护成本、上线周期等方面具有明显优势，在离线使用、移动办公体验方面稍显逊色。由此可见，远程应用模式的移动办公更加适用于高安全要求、快速迁移等场景。

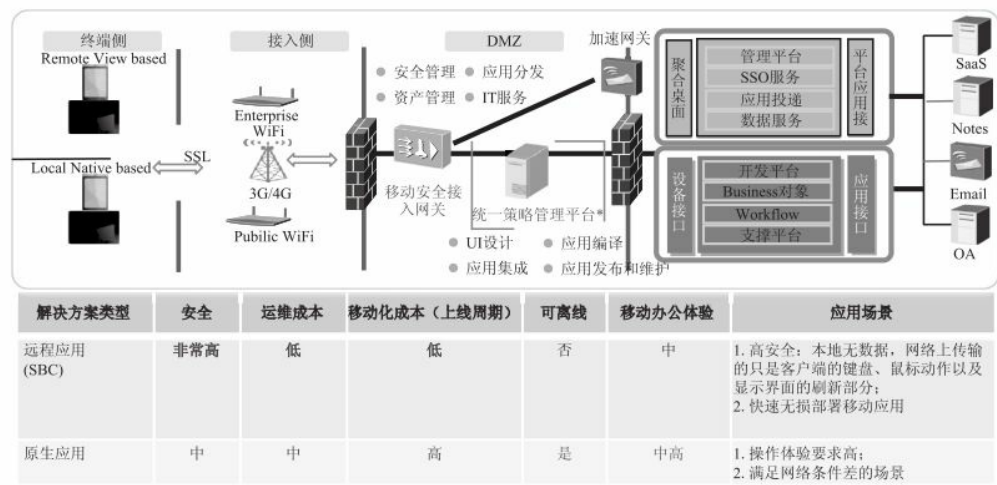


图7-8 主流移动办公方案对比

图7-9 为移动办公方案全景图，图中应用虚拟化（SBC）被公认为移动办公最佳解决方案之一。

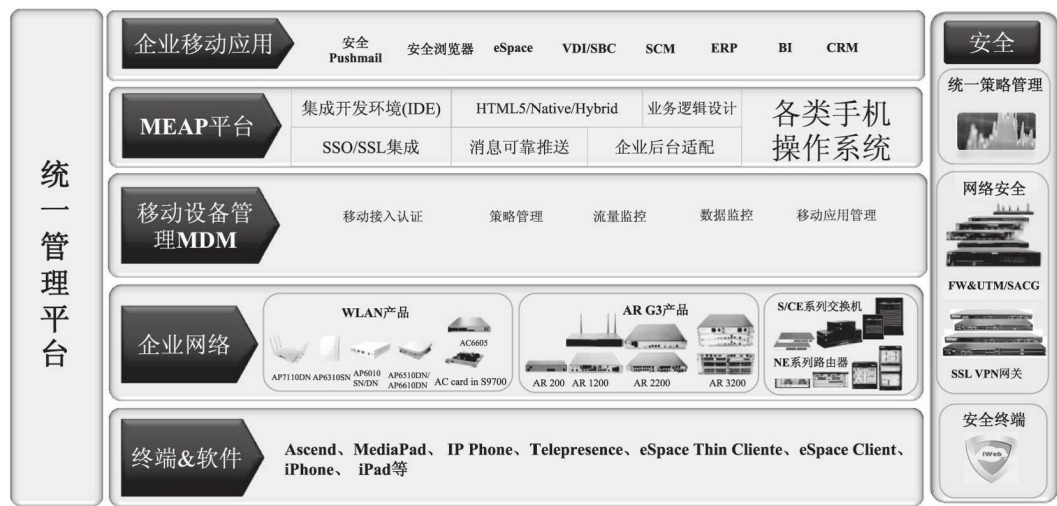


图7-9 移动办公方案全景图

移动办公解决方案（远程应用模式SBC）

图7-10 为SBC移动办公解决方案，向企业用户提供一个应用与数据集

中管理与发布平台，管理员用户可以分别通过管理员Portal进行应用管理、数据管理、用户管理、服务管理。同时，集成商或企业IT部门可以通过开放API进行二次开发。



图7-10 移动办公应用集成平台

7.3.6 移动办公解决方案的特点

移动办公解决方案具有的特点如下所示。

端到端数据安全防护，使企业更聚集于业务发展

- 文档在线编辑，无法保存到客户端本地；若操作超时，则会锁屏。
- 链路层端到端加密，保障传输安全。
- 用户鉴权。
- 企业资源访问控制。
- 文档与应用分离。
- 应用进程间隔离。
- 应用层软件网关。

一键式安装部署

- 无人值守安装，深度定制Linux，界面友好，多节点复用统一镜像。

- 配置简单，自动时间同步，支持远程配置、远程定位。

- 移动应用快速上线，后台无需修改。

- 移动应用快速升级，客户端无需升级。

出众的用户体验

- 定制化的快捷工具栏。

- 拍照插入文档。

- 触摸可编辑放大镜。

- 自动弹出键盘、自动滚屏。

- 分区域滚屏。

- 便捷的任务管理。

资源调度最优化

- 网关负载均衡。

- 应用服务器负载均衡。

通过广域网加速技术，提升应用访问速度

- TCP协议优化。

- 实时流压缩。

- 绘图指令重定向。

- 数据缓存。
- 云模式+广域网加速技术大幅提升应用访问速度。

7.4 云接入的关键技术

7.4.1 桌面云协议简介

在目前云接入领域的关键技术主要是桌面云的接入协议技术，目前业界知名的有微软的RDP协议、思杰的ICA、红帽的Spice协议、华为HDP协议（我们指的接入协议关键技术并非仅指通信协议本身，还包含协议服务器端的实现与客户端的实现）。桌面协议包括具体的远程显示、远程控制、远程音频、远程外设等关键技术，而这些技术的实现具有很大的难度，所以我们认为桌面云协议是云接入最为关键的技术。

7.4.2 桌面云协议关键技术：高效远程显示

从表面上来看，桌面云高效远程显示为一个较为简单的技术，通过操作系统接口来抓取屏幕内容，再经过一定的压缩处理即可在客户端显示服务器端的屏幕内容。例如我们常用的VNC（Virtual Network Computing）就属于这一类型的实现，VNC也具有有一些手段降低带宽，但是我们发现，如果与Citrix ICA、Microsoft RDP进行比较，VNC在带宽方面的劣势非常明显。

这些高性能的桌面云协议中屏幕显示基于什么原理？他们的实现有什么不同？事实上他们在实现的架构上比较类似，都是基于普通计算机的显示原理。

注：由于目前桌面云主要应用的操作系统基本都是微软的Windows操作系统，本文将直接描述Windows平台的实现（见图7-11）。

我们需要远程看到显示的内容，就好像将机器的显示器拉到远端一样。从图7-12中我们可以看到操作系统的软件层次通过操作系统，可以完全获取到显示内容以及和硬件交互的为“Windows显示驱动程序”。如果将“Windows显示驱动程序”发往显卡的数据传输至远程瘦终端的显卡上，即可以达到远程显示的效果。

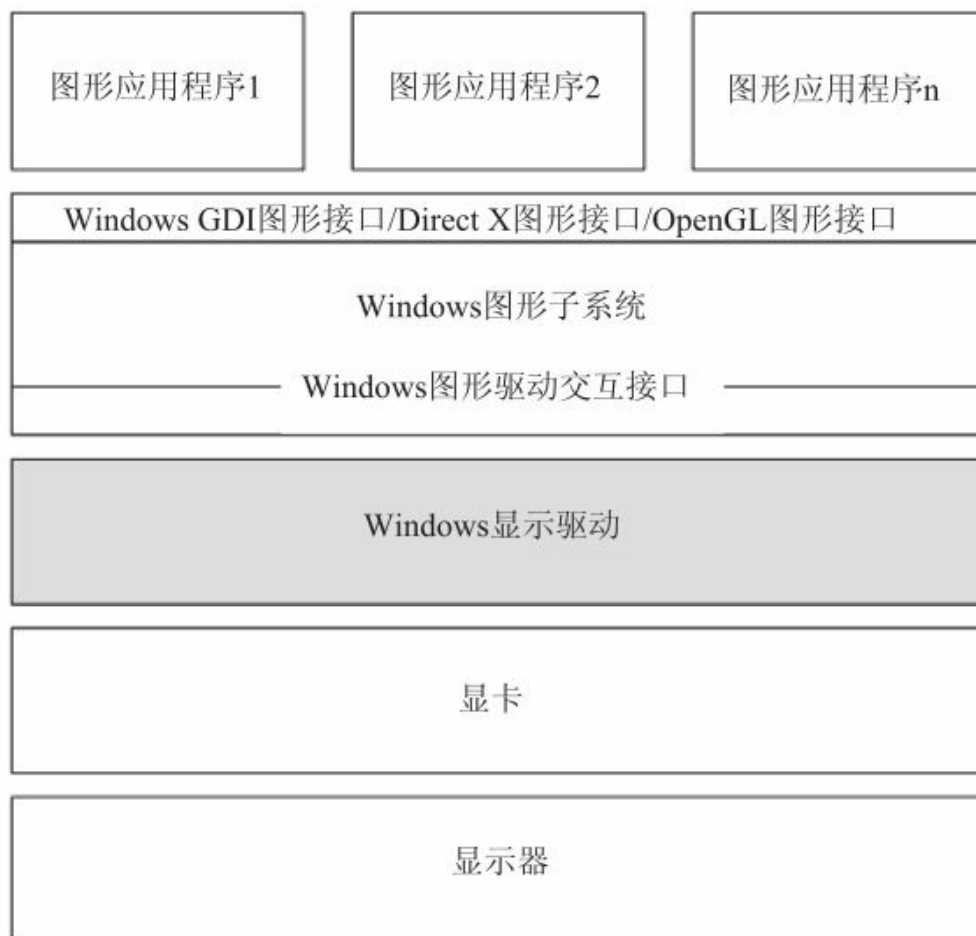


图7-11 计算机屏幕显示原理

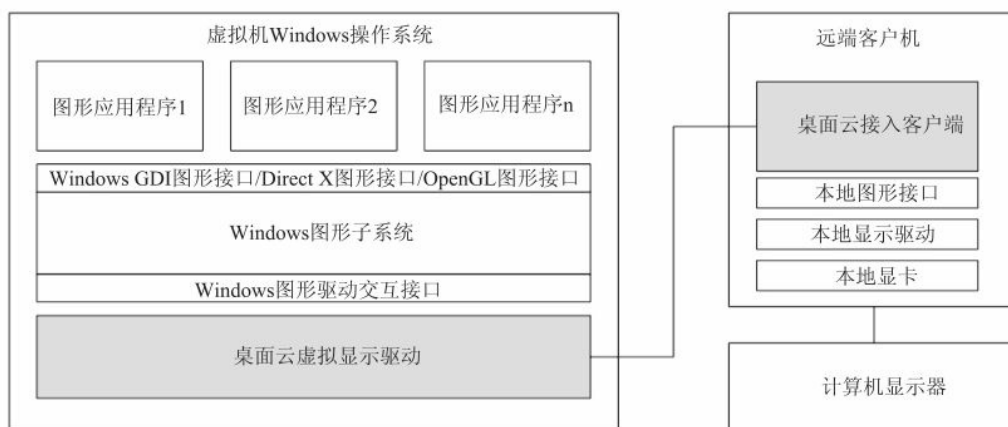


图7-12 桌面云内容在远端客户端显示的原理

目前，业界的实现通常会为运行在虚拟化平台中的虚拟机安装一个远程

虚拟显示驱动，通过虚拟显示驱动来高性能地获取显示的图形指令数据，并将这些数据传送到远程客户机进行显示。

通常应用程序会通过Windows平台提供接口来绘图，这些图形接口调用会通过Windows图形子系统的转换来调用到虚拟显示驱动中（这些图形接口调用我们暂且称为图形指令），图形指令内部的参数描述了图形程序的具体显示，这些数据可以被传输到远程客户端进行重新绘制显示。

这里有两个问题。

为什么不使用Windows平台的接口来直接获取整个屏幕数据，压缩后传输到客户端进行显示，而是要实现一个难度更高、更复杂的虚拟显示驱动来获取图形指令来进行处理？

通常情况下，显卡与计算机上的接口为PCIE，PCIE的带宽非常之高，16X的显卡可以拥有16GB/s巨大带宽，而我们客户端的网卡一般就100MBPS~1000MBPS，实际带宽更少，如果远程客户机与服务器端的距离较远的话，平均每用户的带宽可能不到1MBPS，这样的网络情况与原始的显卡PCIE差距很大，怎样实现远程显示？

这两个问题的答案会在下面进行解答。

（1）2D基本图形显示桌面

大多数的图形应用程序都是2D基本图形显示程序，如Word、Excel、Outlook、Notepad、杀毒软件等，通常我们普通办公场景下的程序都为2D图形程序。所以目前桌面云主要的场景为2D显示场景，如果显示驱动仅仅支持2D显示，遵从微软的显示驱动架构，可以实现微软定义的XPDM（Windows XP display driver model）显示驱动来满足需要。

XPDM架构是微软为Windows Vista之前的Windows版本定义的显示驱动（Vista、WIN7可兼容），目前在桌面云场景下协议服务器端一般也会实现一个XPDM驱动，即图中显示驱动与视频微型端口驱动两部分，该驱动并非用来驱动本地显卡，而是获取Windows图形引擎向显卡发送的图形指令数据，并传输到客户端进行重新渲染显示。通常在桌面协议里面我们仅用XPDM驱动方式来支持2D应用，对于需要3D加速的应用无能为力（可以用软件实现的3D渲染来辅助支持3D应用，但是性能有限）。服务器端获取的这些图形指令需要的数据量非常大，100M网卡

是不够的，所以需要加入许多优化的处理，通常的优化手段具体如下。

- 图像数据压缩：利用各种图像压缩算法对图像内容进行有损或者无损的压缩来降低带宽。
- 指令合并：图形指令的数量有时候会非常之多，通过合并技术可以显著降低指令数量与总体数据量。
- 缓存：通过缓存技术减少服务器与客户端之间的冗余数据交互。

当然实际厂商的技术可能会有不同，且优化手段更多，通过这些类型的优化手段，可以将网络带宽降低百倍甚至更多。

（2）高性能图形支持

高性能图形指的是需要显卡辅助来支持的程序，通常是DirectX或者OpenGL程序。在目前的桌面云场景下一般仅支持普通办公，没有大量部署高性能图形的能力，主要原因是桌面云的部署还处在初始阶段，企业未大量更新到桌面云，高性能图形的桌面云虚拟机成本也更高，导致桌面云总体的技术研发投入有限，在普通办公场景下的技术基本成熟，但是高性能图形方面的高级技术还有待进一步发展，最近各厂商加大了研发投入，如VMware的vSGA（Virtual Shared Graphics Acceleration）、Citrix的OpenGL加速组件、NVidia的VGX等GPU虚拟化技术都已经推出。当然目前在高性能图形方面，直通方式更加通用，拥有更好的兼容性与性能，但是成本较高。

GPU直通实现方式一般如图7-13所示。

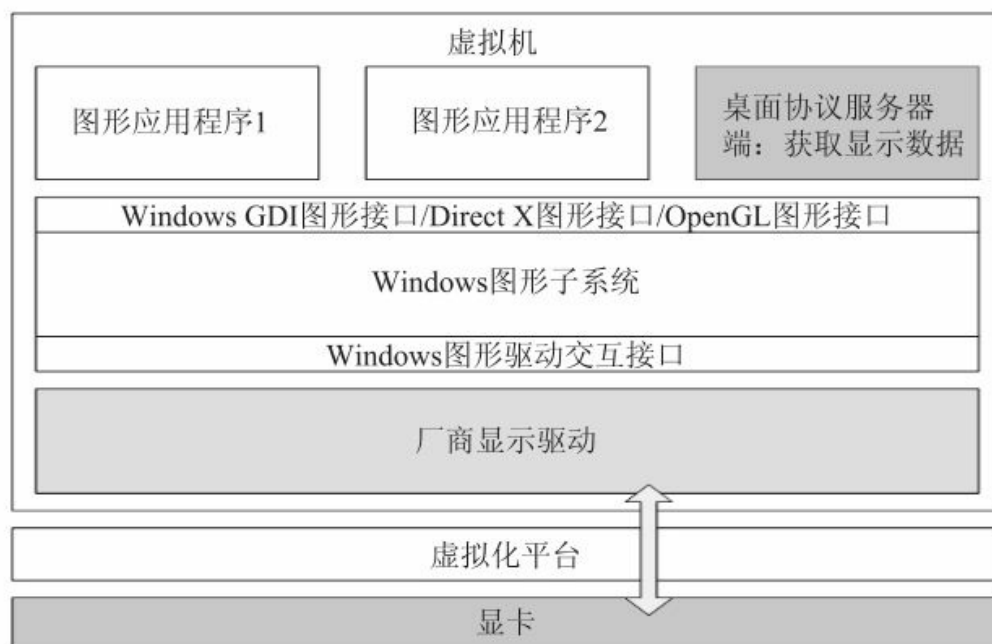


图7-13 GPU直通实现原理

通过虚拟化平台的直通技术可以将显卡直接给虚拟机使用，与物理机接入显卡的效果基本一致，在虚拟机上只要安装了对应显卡的显示驱动，显卡就可以为这个虚拟机提供高性能图形的能力。桌面云服务器端程序将捕获桌面图像数据，来支持远程客户端的显示。这个方式的桌面图像处理方式与前面介绍的2D桌面处理方式有些不同，具体细节不进行介绍。

简单来说，GPU虚拟化/共享能够将一个物理存在的显卡分享给多个虚拟机使用，每个虚拟机将获得高性能图形处理的能力。

前面简单地介绍了一下2D桌面支持时经常使用XPDM方式显示驱动架构来实现桌面云图像的处理，但是实际上微软在Windows Vista以后采用了新的显示驱动架构WDDM（Windows Display Driver Model），如图7-14所示。

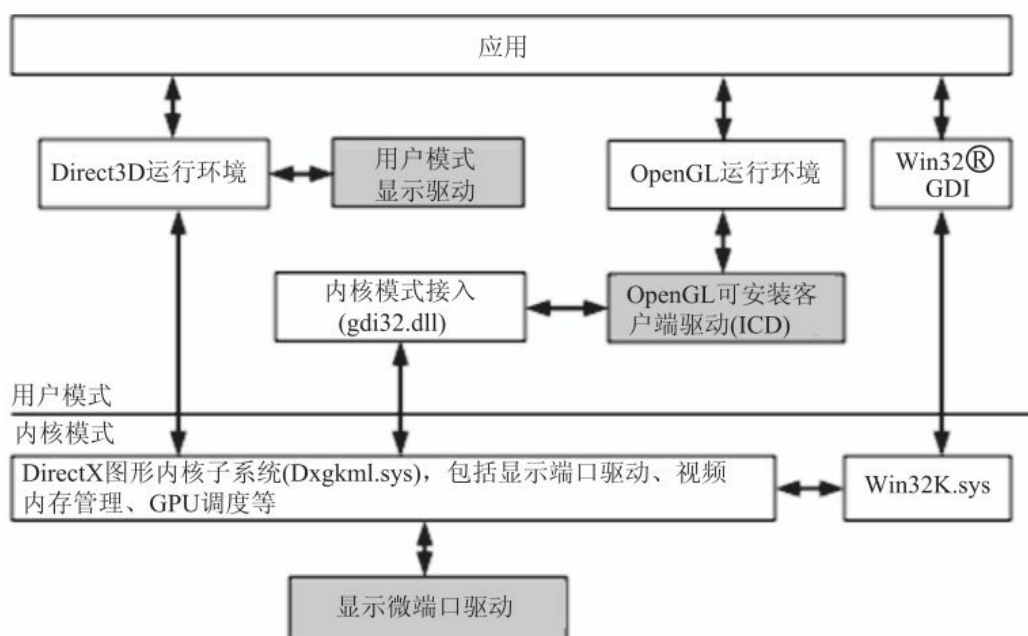


图7-14 Windows Vista、Win7显示架构

WDDM显示驱动架构有三部分，为DirectX服务的用户态模式驱动、为OpenGL接口服务的OpenGL ICD驱动（由于微软主推DirectX接口，OpenGL只是可选组件）以及Display Miniport Driver（工作在内核态负责与硬件交互）。

如果需要支持GPU虚拟化技术，通常需要实现一个虚拟的WDDM显示驱动，来获取Windows对显示驱动产生的接口调用数据，并将这些接口调用重定向到一个GPU共享组件上来进行处理，该GPU共享组件一般会存在于虚拟化平台内部（也有可能是存在于其他处）。处理后将得到渲染后的桌面图像数据，这些数据将被桌面服务器端通过处理发送给客户端，可能是直接将图像编码为H.264码流，也可能是其他图像编码方式（见图7-15）。

为什么不采用与2D图形处理方式一致的图形指令重定向方式将图形指令重定向到客户端进行渲染显示，仅在客户端安装一个相对低性能的显卡？因为2D图形指令进行一系列的优化处理，带宽可以呈100倍以上地降低，仅仅1MBPS以内的带宽即可以满足普通办公的场景需求，但是采用3D应用去支持则难度很大，目前还没有技术可以将3D图形指令重定向带宽降低到如此低的程度。

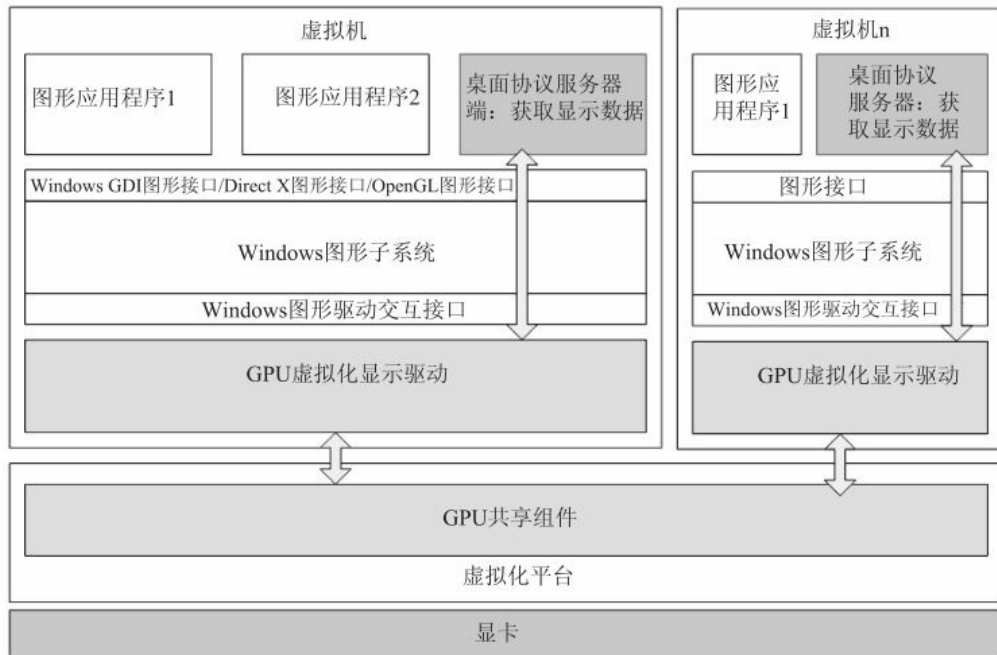


图7-15 GPU技术

7.4.3 桌面云协议关键技术：低资源消耗的多媒体视频

多媒体视频播放器对桌面云来说是一个挑战，如用户体验、音视频同步、带宽等。目前在桌面云支持多媒体视频一般有两种方式，一种是将服务器端的多媒体视频播放器的图像进行重新视频编码，将视频编码传输到客户端进行解码显示；另外一种为视频重定向方式，一般通过捕获播放器需要播放的视频编码流，直接将视频编码流发送到客户端进行解码显示。很明显，第二种视频重定向方式看上去效率更高，服务器少了视频解码与重新编码的资源消耗，但是实际上这种方式非常受限，无法得到广泛的支持（见图7-16）。

第一种方式由于在桌面虚拟机中的播放器将视频进行了解码，这里会有较大的解码CPU资源消耗，在对视频区域进行编码时消耗更大，这样将降低一个服务器能够支持虚拟机数量的密度。另外，对视频区域的识别也是一个重要的技术点，通常会通过识别刷新频率超过一定帧率的图像变更区域来识别。

第二种方式由于仅仅在服务器端截获待解码的视频码流并传输到客户端进行解码显示，对服务器端的开销较小，目前比较流行的技术为针对MediaPlayer支持的多媒体重定向技术。但是该技术在国内的实用性并不

高，毕竟国内很少使用MediaPlayer播放多媒体文件，所以第二种方式实际比较受限。当然技术也在发展，对其他播放器能够支持的多媒体重定向技术相信后续也会出现，这将降低对服务器端的资源消耗。



图7-16 服务器端解码视频播放（左）与多媒体重定向（右）原理

7.4.4 桌面云协议关键技术：低时延音频

桌面云协议对音频的支持与前面介绍的2D图形支持实现比较类似（见图7-17）。

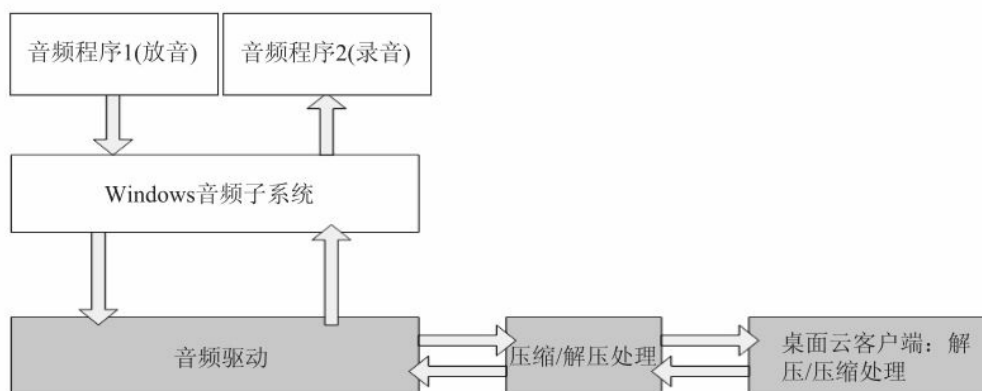


图7-17 桌面云音频输入输出实现原理

通常桌面协议服务器端可以在虚拟机里面实现一个音频驱动，音频驱动会与Windows的音频子系统（音频引擎）进行交互。在放音阶段，音频驱动将收到Windows音频子系统发送过来的音频数据，经过压缩处理后传输到桌面云客户端，客户端进行解码并进行放音。在录音阶段，客户端将获取客户端本地的录音数据，并将数据进行压缩后传输到服务器端，服务器端进行解码后由音频驱动返回给Windows音频子系统。由于音频对延时非常敏感，整个过程要关注对延时的控制。

7.4.5 桌面云协议关键技术：兼容多种外设

在通用的系统上，常用的外设种类有USB外设、并口外设、串口外设等，目前来看USB外设占据主流，解决USB外设的支持即可满足目前最为流行的外设硬件支持。

实现该部分关键技术需要先认识目前传统USB外设工作的原理（见图7-18）。

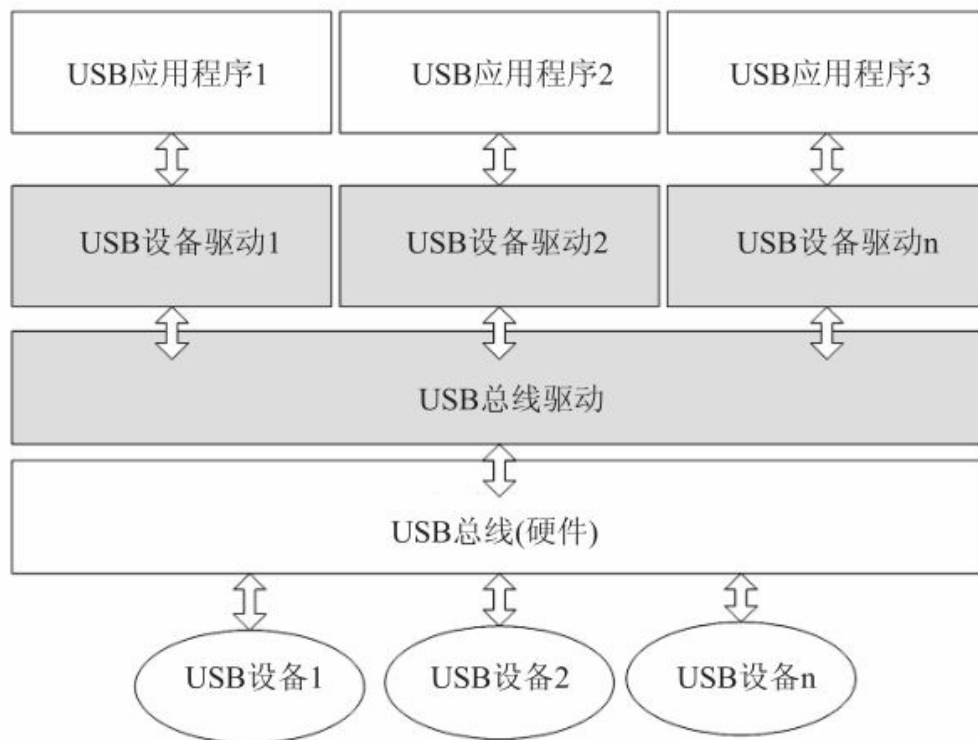


图7-18 USB实现原理示意图

从图7-19可以了解到，所有USB外设的正常工作在软件层面依赖的是USB总线驱动。一个应用需要使用USB外设必须与USB设备驱动进行交互，而设备驱动的工作完全依赖USB总线驱动来交互USB设备数据，与硬件的交互都是由总线驱动来代理完成的。从我们的理解来看，从USB总线驱动入手是软件层面最合适的方式，将USB总线驱动与本地硬件的交互远程化，转化为本地USB总线驱动与远程客户机USB硬件总线的交互（见图7-19）。

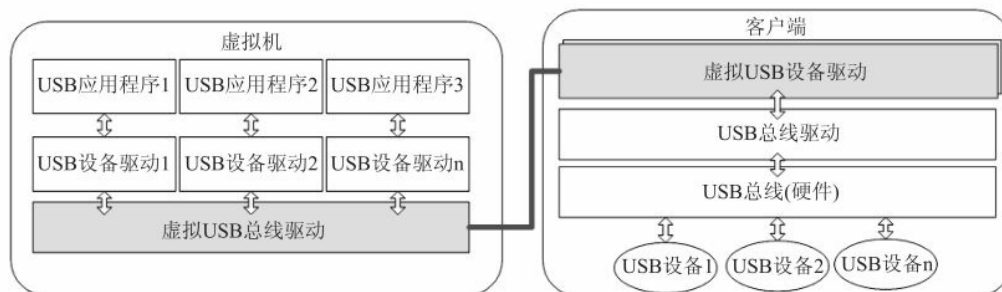


图7-19 桌面云USB外设支持原理

通过在虚拟机内部实现虚拟USB总线驱动可以与客户端的硬件设备进行

通信交互，交互也不是直接的通信，而需要在客户端上开发一个虚拟USB设备驱动，通过虚拟化USB设备驱动与客户机的USB总线驱动进行交互。当有一个设备插入时，客户机的USB总线会发现一个新设备插入，此时将启动一份虚拟化USB设备驱动的实例，如果有多个设备需要同时被重定向时，需要多份虚拟USB设备驱动实例运行在客户端上。而设备对应的真实USB设备驱动安装并运行在虚拟机中，与虚拟机USB总线驱动进行交互，这样对虚拟机中的USB设备驱动来说并没有太大感知，对应用程序也没有太大感知，原因是，远程的这种数据交互会带来延时，有些设备驱动在设计的时候考虑了一些超时的处理。

这种方式表面上来说能够很好地支持各种USB外设，但是实际上来说也有可能存在问题，一是很难非常好地做到对设备的兼容，二是一些设备重定向后带宽非常之大而无法被使用，所以一些设备无法正常地使用USB总线方式来实现设备的重定向，比如摄像头，如果走总线重定向的方式，带宽有数十兆，甚至更多，这基本无法实际部署。针对这一类型的设备，一般会单独为它优化来使其可以满足实际商用。如针对摄像头，我们可以采用如下方式实现优化，如图7-20所示。

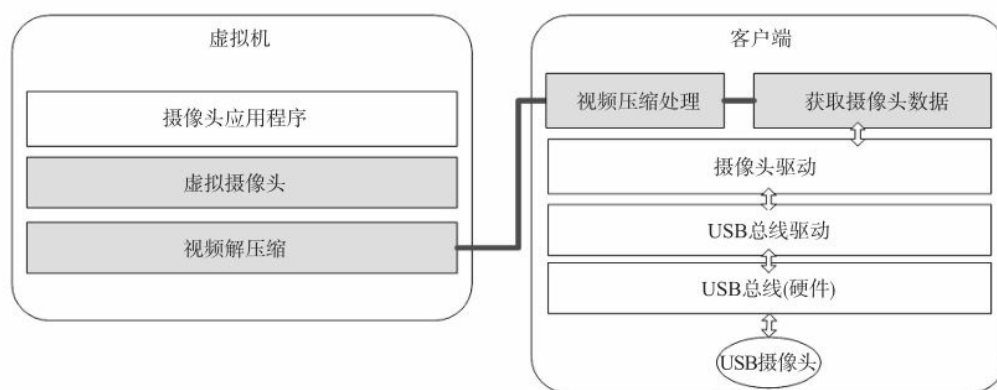


图7-20 摄像头外设支持原理

我们可以在客户端通过应用程序级别的接口来获取摄像头的的数据（一般为位图数据或者YUV数据），再将数据通过视频压缩算法（如H.264）进行压缩处理，发送到服务器端，服务器端解码摄像头视频数据后通过虚拟摄像头提供给应用程序使用。有了视频压缩技术支持，这种基于摄像头的重定向技术比基于USB总线的重定向技术带宽下降数十倍。除了摄像头类型的设备，其他类型的设备也有可能进行特定的重定向处理，这取决于单独实现针对这类型设备重定向的价值。

7.4.6 桌面云协议总结与其他实现

前面介绍了桌面云协议的一些重要的关键技术实现，这些技术主要运行于虚拟机内部，包括显示（2D、3D、多媒体）、音频、USB外设、键鼠，除了这些还存在其他的一些关键技术，另外目前桌面云的技术也在发展，肯定会推出新的技术（见图7-21）。

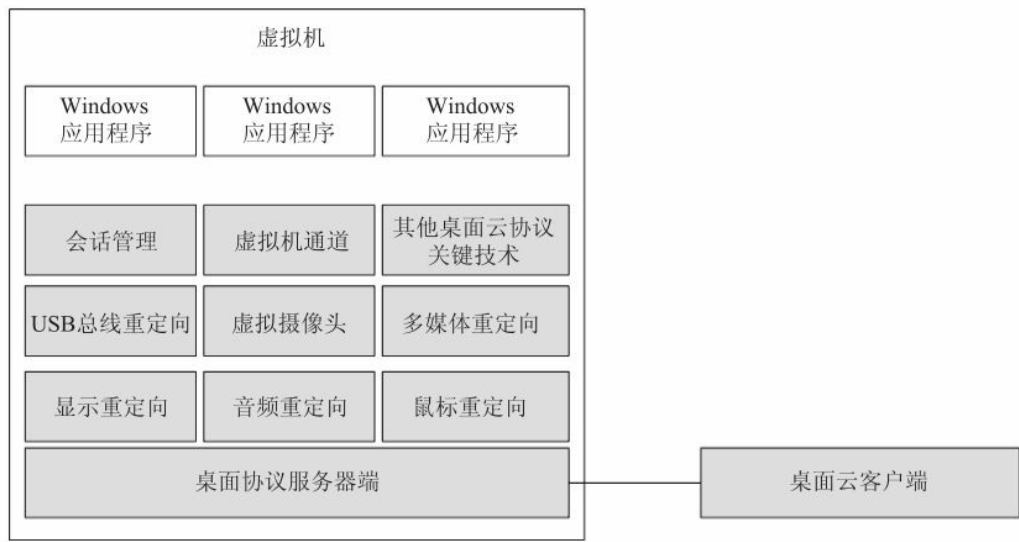


图7-21 通常桌面云技术实现原理示意图

目前的关键技术实现都是运行于虚拟机内部的，也就是说这些技术与虚拟化平台没有关系，通常桌面云协议服务器端也可以安装于物理机器上（例如RDP就可以远程连接虚拟机或者物理机）。这个实现架构有与虚拟化平台或者硬件平台无关的优势，但是它也有缺点，如整体实现完全需要基于操作系统来实现，支持Windows的实现与支持Linux的实现将存在很大的不同。当然目前Windows是绝对主流，所以各桌面云商业厂商主要都在支持Windows操作系统。红帽的Spice采用了另外一套实现架构，这种实现架构能够做到绝大部分与操作系统无关，也就是说它可以更好地支持多种操作系统（见图7-22）。

Spice的架构中大部分的实现都在虚拟化层，而不是在虚拟机中，所以Spice提供了对Linux桌面云的支持。为什么Spice在虚拟机中还提供了一个虚拟显示驱动？原因是如果不提供显示驱动，它的性能会很差，就好像如果你有一台电脑安装了显卡，但是不安装显卡对应的驱动，实际上无法使用这个显卡，同样如果没有显示驱动的加速Spice虚拟显卡，仅仅工作在VGA的模式下，则无法获取上文中提到的图形指令，无法高性

能地处理重定向图形显示。当然Spice的架构还有其他独特的优势，由于工作在虚拟硬件上，不同于操作系统内部，它还可以不依赖虚拟机内部的网络与客户端进行通信，这样可以防止很多用户自己对网络进行更改，造成桌面云主机无法使用的情况，它还可以在开机启动过程中看到整个系统启动的过程（虚拟机内部的桌面协议要等待操作系统启动后才能连接），这些都是Spice的优势。当然Spice的方式也有一些明显的缺点，它和KVM虚拟化平台强绑定，目前官方无法支持其他虚拟化平台，无法支持物理主机接入，由于它的主要实现实体在虚拟硬件层，一些针对Windows的桌面协议优化无法支持（或者支持付出的代价更大），如多媒体重定向、摄像头重定向等，也无法像RDP一样可以支持Windows Server操作系统多用户的接入。两种实现方式各有优缺点，目前来看主流实现以基于Windows操作系统层实现的桌面协议为主。

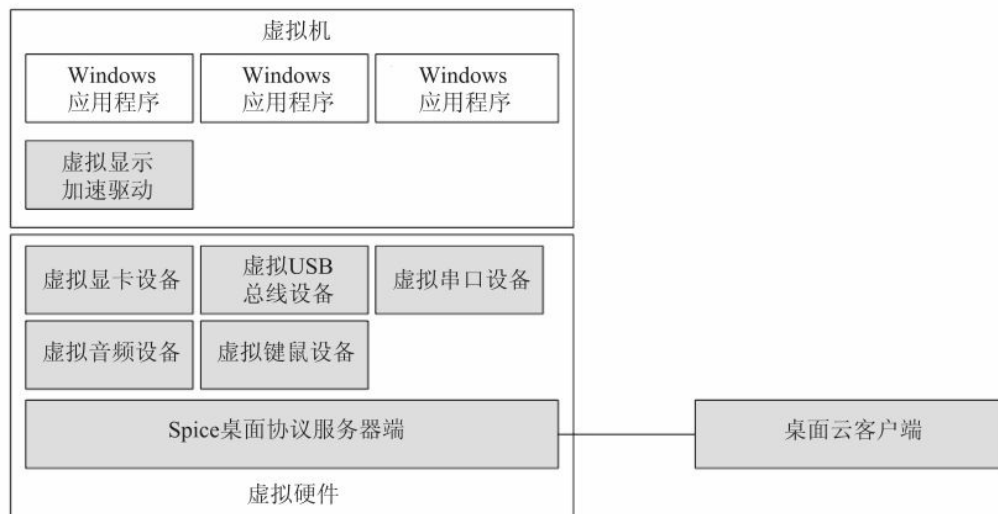


图7-22 Spice实现原理示意图

无论是基于Windows操作系统层实现各种驱动，还是基于Spice在虚拟硬件层次实现各种虚拟硬件，其实都是希望在底层将原本属于本地的交互转换为远程的交互，而屏幕显示的效果与本地交互相同。这种转换涉及显示重定向-用户视觉、音频重定向-用户听觉、键鼠重定向-用户触觉、外设重定向-各种外设使用，目的是利用一个简单的桌面云终端替代PC物理机为用户服务，并且需要非常高的用户体验，包括用户操作延时、桌面云显示质量、视频质量与流畅度、音频延时与质量、传输带宽大小等。整个桌面云的实现技术涉及类别非常多，包括各种虚拟驱动/虚拟硬件（显示、音频、键鼠、USB）、各种图形算法、音频算法、视频算法、带宽优化、数据去重等，所以桌面云协议的技术难度较大、门槛较

高，可提供商用解决方案的也仅仅几家公司。

7.5 云接入的发展趋势

7.5.1 云接入的未来发展

如图7-23所示，云接入未来在云端提供各种桌面、应用、数据，统一空间：

- 各种设备无缝接入；
- 跨平台提供一致的用户体验；
- 企业应用商店及统一门户，聚合各种“应用+数据+桌面”。

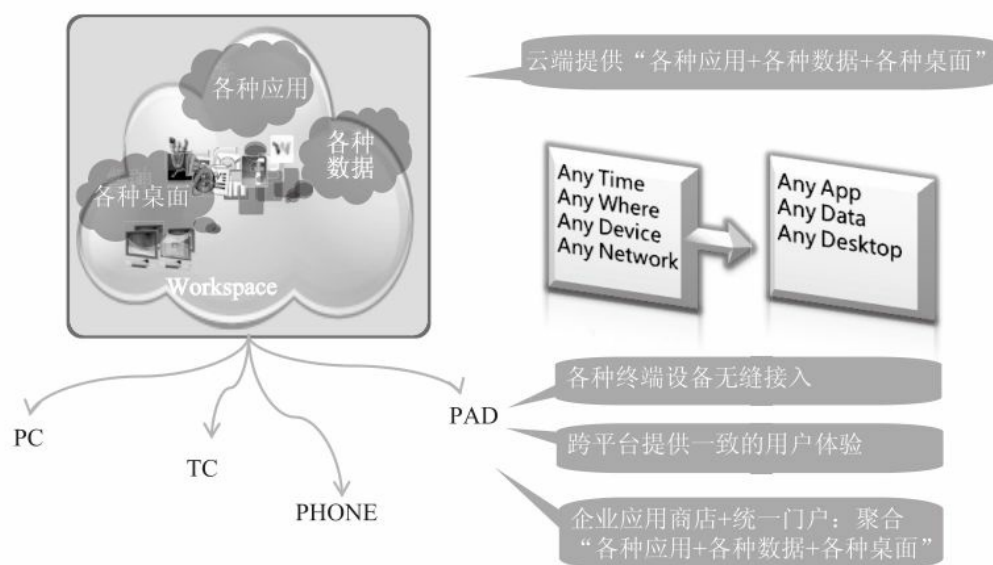


图7-23 云接入未来发展

平台开放化

作为基础平台，封闭架构带来不兼容性，无法支持异构虚拟机系统，也难以支撑开放合作的产业链需求。随着云计算时代的来临，桌面虚拟化管理平台逐步走向开放平台架构，多种厂家的虚拟机可以在开放的平台架构下共存，不同的应用厂商可以基于开放平台架构不断地丰富云应

用。

连接协议标准化

桌面虚拟化连接协议目前有VMware的PCoIP、Citrix的ICA、微软的RDP、NComputing的UXP等。多种连接协议在公有桌面云情况下，将带来终端兼容性的复杂化，终端将需要支持多种虚拟化客户端软件，对于嵌入式的云终端来说，限制了客户采购的选择性和替代性。

未来桌面连接协议标准化之后，将解决终端和云平台之间的广泛兼容性，形成良性的产业链结构。

虚拟化客户端硬件化

当前的桌面虚拟化和应用虚拟化技术对于富媒体的客户体验和传统的PC终端相比还是有一定的差距的，主要原因是对于2D/3D/视频/Flash等富媒体缺少硬件辅助虚拟化支持。

随着虚拟化技术越来越成熟及广泛应用，终端芯片将可能逐步加强对于虚拟化的支持，从而通过硬件辅助处理来提升富媒体的用户体验。特别是对于PAD、智能手机等移动终端设备，如果对虚拟化指令有较好的硬件辅助支持，将促进虚拟化技术在移动终端的落地。

7.5.2 VDI

据IDC预测，桌面云这个市场的潜力是非常大的。随着人们对桌面虚拟化好处认知的提高，以及对桌面虚拟化的需求的提出，相关技术的不断完善，桌面虚拟化必将普及。虽然面临的问题很多，但并不是说桌面虚拟化将就此止步，还没有哪种技术是不存在潜在缺陷甚至陷阱的。当人们有这个桌面云需求时，一切问题都不再成为问题。现在人们对虚拟化已经有了需求，而且这个需求是不断深化的，目前已经有不少企业部署了VDI。有了需求就有了市场，需求能促进技术的进步，桌面虚拟化的普及仅仅是一个时间问题。桌面云的前景是非常乐观的。

存储的技术发展，可以降低整体成本

若VDI要保持1:1的磁盘镜像比例，会让传统服务器存储的成本高昂得让人止步。这也是为什么VDI部署关注非持续性的“共享”镜像，以及为

什么VDI采用仍然处于边缘地带。现在，很多厂商都提供块级别、单个实例主存储。使用重复数据删除技术可以大大节约整体成本。举个例子，5000个虚拟桌面镜像，完整复制40G大小加上一个2GB的交换空间总共需要210TB的存储容量。应用存储数据删除技术，它被压缩到小于4TB，一个4TB I/O卸载引擎可以支持5000个桌面。这意味着现在可以创建拥有上百个用户的VDI环境，每个用户有一个专门的磁盘镜像，而只需要以前购买一个共享镜像系统的价格。

图形性能领域的技术发展，用户应用体验达到PC的水平

另一个重要的突破是图形性能领域。多个供应商现在为远程VDI主机服务器提供插入式GPU卡，可以实现卸载进程，并对远程协议显示数据流进行编码与压缩。为服务器添加GPU卡意味着可以将所有的处理能力贡献给图形体验，无需加重现有CPU压力而牺牲用户体验。这些插卡也允许GPU虚拟化，这将让诸如3D CAD、Photoshop与视频编辑应用能够通过VDI进行访问。

VDI：在你身边的桌面

这些存储与图形功能的改进意味着虚拟桌面基础架构现在的使用范围比之前大得多。这能从部署客户部署桌面的数量（拥有1万个以上）看出来。

首先说明，笔者不是建议桌面的未来就是VDI，或者所有的Windows桌面应该迁移到VDI。在过去几年里，我们可能看见VDI在企业桌面里的市份额只有5%~10%。现在VDI可以支持1：1持久的磁盘映像和图形密集应用，VDI将在企业桌面中占据40%~50%的份额。

7.5.3 DaaS

DaaS是什么？其指将Windows桌面和应用以云服务的形式向用户交付。它具有如下特性，如表7-1所示。

表7-1 DaaS特性

特性	说明
----	----

高可用性	7×24的服务保障
可移动性	用户可以通过互联网随时随地接入桌面和数据
安全	定时数据备份集中管理IT架构保证企业数据安全重要数据零丢失保证
灵活升级扩容	无需考虑软件升级新用户扩容短时间完成
IT费用节省	IT预算可预期无软硬件投资费用SP提供专业服务和支 持

IT服务商部署DaaS的主要驱动力为以下几点：

- 巨大的潜在市场，提供差异化服务，新的盈利点；
- 增强用户粘性；
- 带动其他服务（备份/防病毒/企业应用托管/文件存储/数据迁移）；
- 现有的PC外包市场大约300亿美元（数据来源：Gartner）。

为什么需要DaaS？

- 更佳的桌面体验
 - 不同的设备，一样的桌面体验。
 - 无需关机，随时随地进入工作状态。
 - 统一部署和升级应用软件，更快捷。
- 简单易用的虚拟桌面
 - 自助申请，自动部署。
 - 在线支付，立即可用。
 - 专业级的运营服务。
- 投资可以预期

- 初始投资成本低：按需申请，用完即退。
- 运维成本低：无需自己运维。
- 终端成本低：TC更换周期长，故障率低。

➤ 数据集中管控更安全

- 数据集中管控，接入终端无保密数据。
- 企业租户安全隔离，确保网络安全。
- 高效快捷的外设管控策略和审计机制。

DaaS的典型场景

托管（hosting）模式，适用于政府、园区。

混合云模式，适用于大型企业/垂直行业。

公有云服务模式，适用于中小企业/分支机构。

7.5.4 移动办公

伴随着智能设备以及3G、4G网络的蓬勃发展，企业移动办公方式向3A（Anyone Anywhere through Any Network Use Any Device to Do Anything）发展的趋势已经十分明显，大量企业纷纷升级IT基础设置，提升办公效率。各种技术方案层出不穷，各领风骚。

据某公司提供的一项调查报告显示，预计到2020年，全球89%的企业都将采用移动办公的工作方式，而目前有24%的受访者表示已经全面部署了移动办公，另有38%将扩大移动办公的部署规模，还有21%的受访者表示预计在未来两年内实现移动办公。这也就意味着移动办公市场将从2012年的24%迅猛上升到2014年的83%，年复合增长率高达86%。

据这份报告显示，北美地区的移动办公市场较为领先，90%的企业表示已经部署或正在扩大部署移动办公解决方案，紧随其后的是中国（85%），之后则是巴西、印度、英国、法国和德国，其百分比都超过了70%。

这些数据显示移动互联网时代的到来。移动办公为何受到这么多国家或地区用户的青睐呢？此报告也罗列了原因，73%的受调查者表示移动办

公方式能够使工作方式更加灵活，并有53%的受调查者认为这让出差或者旅游更加方便，48%的受访者表示能够降低办公场所的费用。吸引（47%）和留住（44%）人才也是企业部署移动办公解决方案的驱动因素之一。

移动办公一直被认为是最理想的工作方式，企业对员工能够提供诸多方便：第一，移动办公能够提供比较弹性的办公环境；第二，能够降低企业员工的出行、生活成本；第三，能降低企业办公场所成本；第四，还能吸引更多、更好的人才。过去人们常说的一句话——21世纪最缺的是人才。如果企业实现了移动办公，在员工不方便到办公室上班的时候可以允许员工在家移动办公，既保证了业务的连续性，又能提供人文关怀，这样的公司谁不想留下呢？

那么，移动办公什么时候才能实现呢？某公司授权一家市场调查公司独立做了一项针对移动办公未来趋势影响的调研，涉及19个国家、2000名IT高管。根据这项移动办公调查报告显示，移动办公已经成为全球企业办公的重要趋势，调查中大约24%的企业已经部署移动办公方案，21%的企业计划在两年内部署移动办公方案，预计到2014年，将有86%的企业实现移动办公。预计到2020年，全球企业办公空间将减少17%，中国企业的办公空间将减少21%。预计每10人平均需要7张办公桌，在新加坡、荷兰、美国、英国平均需要6张办公桌，而在中国，这个数字是7.69。

通过被访者对可能会进行移动办公的地点的回答，可以一窥未来工作场所的特点。在被访者回答中，95%的中国企业用户会选择在本地办公室接入移动办公平台，亚太地区平均数据为73%；90%的中国企业用户会选择从家里接入移动办公平台，亚太地区平均数据为62%；84%的中国企业用户会选择从公司的项目所在地或其他现场接入移动办公平台，亚太地区平均数据为64%；81%的中国企业用户会选择从客户、伙伴、供应商那里或活动现场接入移动办公平台，亚太地区平均数据为53%；48%的中国企业用户会选择从咖啡厅、餐厅、图书馆以及类似公共区域接入移动办公平台，亚太地区平均数据则为30%；其余地点还包括在各种交通工具上。如此看来，移动办公使得办公并不限于办公室或者家中，而是真正实现了随时随地办公。

在调查中，中国企业用户对于移动办公的热情远高于其他国家和地区，考虑到中国地缘广大、东西部发展速度不均衡，在沿海地区和一、二级城市的企业用户的移动办公水平相当乐观。44%的中国被访企业表示有

专门的团队进行移动办公解决方案的开发和管理，28%的企业表示将在2012年底之前成立这样的团队。之所以引入移动办公，其主要驱动力是来自员工的主动要求。而在中国，IT高管表示第一驱动力是因为终端设备种类繁多，其次是出于安全和风险控制的考虑。调查显示，全球人均拥有6台不同的计算设备，而在中国，这一数字为3.06台。

第8章 云管理与自动化的关键技术架构与应用

从云计算的功能来看，云管理面向虚拟基础设施的资源自动化发放和一体化的运维管理，可以被部署到一个自助服务、自动化的公有云或私有云操作环境中。其主要功能包括服务门户、服务目录、权限管理、容量管理、IT资源管理（支持异构）、性能管理、配置变更管理、生命周期管理，并且通过自动化引擎编排成一个完整的资源供给管理系统。

随着企业的规模发展，企业IT系统从小规模走向大规模，从单一资源池向多资源池扩充，所以云管理平台除了解决多资源池容量上的管理问题，还需要管理不同的硬件设备和虚拟化资源。

在企业内部，系统管理员给不同的组织管理员开放租户界面，借助云管理平台，业务人员直接在租户界面完成业务的申请和发放。这种模式可以让企业提供IT企业的出租业务。

云管理系统通过开放API，与企业IT管理平台或运营系统对接，被集成到企业IT系统。

随着企业的规模不断扩大，企业不同组织对IT资源会有差异化诉求，如何将虚拟化组织化、差异化，将资源灵活分配给各组织？请看如图8-1所示的云管理的业务模型。

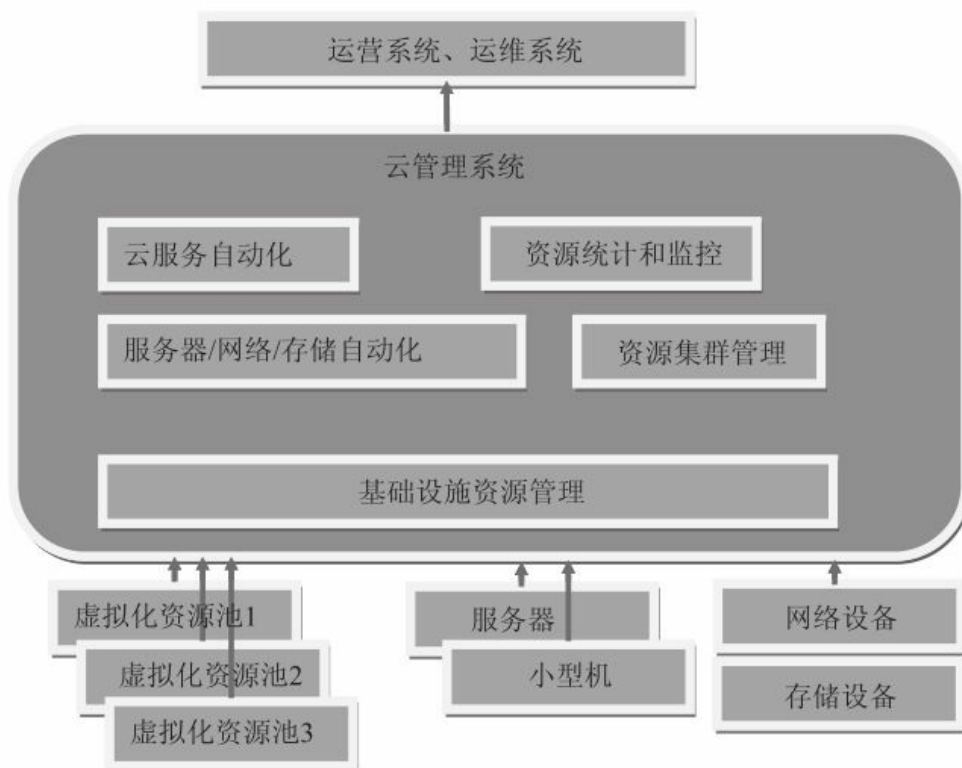


图8-1 云管理平台业务视图

云管理平台对不同的设备和虚拟化资源进行整合管理，上层业务感知不到物理设备的更换和升级以及虚拟化平台的切换。云管理平台提供维护统一入口，可以提升运维效率。通过为每个部门分配独立的虚拟数据中心，使各部门之间的云资源相互独立，使用时互不感知。云管理的主要特点包括以下几点。

➤ **多资源池管理：** 企业从小规模走向大规模，需要云管理提供给管理员统一的资源池管理平台，运行在资源池之上的企业应用感觉不到下层资源池中硬件和虚拟化软件的差异，下层硬件设备的更换升级亦对用户和业务零感知；虚拟化和物理服务器统一管理平台兼容业界主流的虚拟化产品和操作系统，兼容客户现有IT资源；提供基于资源池的统一编排和调度；可实现设备自动发现，资源快速发放，缩短业务上线时间和维护效率。

➤ **从企业内部应用走向多租户的应用：** 软件定义数据中心，网络隔离，通过组织管理，不同部门可以独立使用云资源，共享虚拟化资源，同时网络资源安全隔离，支持分权分域。

➤ 应用弹性伸缩： 简易的应用模板设计工具，给各租户共享应用模板，把业务上线时间缩短到半天，实现应用实例快速部署。支持应用根据资源负荷弹性伸缩，提高设备利用率。

➤ 从封闭走向开放： 提供丰富的二次开发接口，通过应用定制，其他外围系统可通过二次开发包方便地调用平台云计算资源，与平台软硬件设备实时交互，实现丰富的系统集成功能，如与高性能计算业务、程序设计系统等SaaS应用的便捷融合。通过面向下层的标准化接口（南向插件机制），可以快速将新的硬件设备和虚拟化软件纳入云计算资源池，灵活满足企业需求。

8.1 业务应用驱动的拉通计算、存储、网络自动化

随着IT技术的飞速发展，设备更新周期越来越短，服务器的处理能力每年都会增长50%以上。在传统的数据中心，IT部门需要维护大量的服务器、存储、网络设备，同时也要维护应用软件，在应用软件升级、新业务开通时，经常看到IT运维人员通宵达旦地设计业务。

降低维护成本，降低人力成本，缩短业务开发周期，是IT运维人员考虑的重要问题，也是云计算资源池管理平台优先解决的问题。

自动化的发展，使云计算在不到十年的时间普及到普通企业。云管理业务管理中，将计算、存储、网络作为资源配额给企业不同的组织部门，各部门根据业务需要，在组织中发布自己的应用。自动化服务除了提供网络、设备的自动化部署，也需要按需使用，自动调度，动态伸缩。

8.1.1 自动化部署

自动化部署包括软件安装、补丁管理、自动化报表和健康检查、自动扩减容等。

软件安装

除了管理软件本身的产品安装，重点是服务器操作系统及驱动、虚拟化操作系统和应用软件的安装。云管理软件能够实现从应用到服务器的垂

直自动化部署。对于数据库场景，云管理软件也能够支持服务器的裸机管理，实现自动管理和手动部署结合的应用场景。

补丁管理

对管理软件本身，其可实现补丁的自动化部署不会影响业务；对应用软件，其可做到应用软件的自动化推送和更新，提供灵活的软件执行策略给业务管理员执行，可以结合场景在晚上定时执行、立即执行、手动执行等。

自动化报表

管理员自定义报表内容，定期自动导出各类资源的占用和消耗情况，供运维人员对数据中心扩容做参考。

健康检查

自动收集软硬件信息并生成报表，同时对不同资源的健康状态汇总，通知管理员。

自动扩减容

在服务器资源扩容后，能够自动发现设备并扩容集群。

8.1.2 动态调度

云计算系统中有很多主动管理的功能，大部分都是管理员感知不到的，如虚拟机死机检测、休眠检测、虚拟网络的被攻击检测、管理系统本身的故障检测等，这里仅列出管理员常用的主动管理功能，并进行说明。

虚拟机HA

虚拟机HA（High Availability）机制，可提升虚拟机的可用度，允许虚拟机出现故障后能够重新在资源池中自动启动虚拟机。

在已经创建的集群中，如果高级设置中的HA功能已经启用，那么用户在该集群中创建虚拟机时，可以选择是否支持故障重启，即是否支持HA功能。

当物理服务器宕机等引起虚拟机故障时，系统可以将虚拟机迁移到其他物理服务器重新启动，保证虚拟机能够快速恢复。目前系统能够检测到的引起虚拟机故障的原因包括物理硬件故障、系统软件故障。

重新启动的虚拟机，会像物理机一样重新开始引导，加载操作系统，所以之前发生故障时没有保存到硬盘上的内容将丢失。

对于未启用HA功能的虚拟机，当发生故障后，此虚拟机会处于停机状态，用户需要自行操作来启动这台虚拟机。

虚拟机DRS

动态资源调度（DRS）采用智能调度算法，根据系统的负载情况，对资源进行智能调度，达到系统的负载均衡，保证系统良好的用户体验。

动态资源调度策略针对集群（Cluster）设置，可以设置调度阈值、定义策略生效的时间段。在策略生效的时间段内，如果某主机的CPU、内存负载阈值超过调度阈值，系统就会自动迁移一部分虚拟机到其他CPU、内存负载低的主机中，保证主机的CPU、内存负载处于均衡状态。

虚拟机QoS

客户可以自定义必须在同一主机上运行或必须分主机运行的虚拟机，或者限定某些虚拟机只能在部分主机范围内运行和迁移。

虚拟机QoS功能实现了可衡量的计算能力，用来保证虚拟机的计算能力在一定范围内隔离了虚拟机间由于业务变化而导致的计算能力的相互影响，满足了不同业务虚拟机的计算性能要求，同时可以更好地控制计算资源，最大程度地复用资源，降低成本，提高用户满意度。

虚拟机QoS主要体现在CPU QoS和内存QoS。

➤ CPU QoS

虚拟机的CPU QoS用于保证虚拟机的计算资源分配，隔离虚拟机间由于业务不同而导致的计算能力相互影响，满足不同业务对虚拟机计算性能的要求，最大程度地复用资源，降低成本。

创建虚拟机时，可根据虚拟机预期部署业务对CPU的性能要求而指定相应的CPU QoS。不同的CPU QoS代表了虚拟机不同的计算能力。指定CPU QoS的虚拟机，系统对其CPU的QoS保障主要体现在计算能力的最低保障和资源分配的优先级。

➤ 内存QoS

内存QoS提供虚拟机内存智能复用功能。通过内存气泡占用等内存复用技术将物理内存虚拟出更多的虚拟内存供虚拟机使用，每个虚拟机都能完全使用分配的虚拟内存。该功能可最大程度地复用内存资源，提高资源利用率，且保证虚拟机运行时至少可以获取到预留大小的内存，保证业务的可靠运行。

系统管理员可根据用户实际需求设置虚拟机内存预留。内存复用的主要原则是优先使用物理内存。

应用自动扩缩

系统按照用户预先设定的应用资源变更策略，自动调配伸缩组内的计算节点数量，随应用负载的变化而变化。

弹性伸缩是使用云计算的核心收益之一，可以实现计算资源与业务负载之间的动态匹配，不仅可以更好地支撑业务的可用性、改善用户体验，也能最大限度地提升资源的使用效率，有效避免为了支撑将来的业务高峰而预先多分配额外的资源。

用户可以针对系统对不同类型的应用设置不同的资源使用策略。系统分为三种策略类型，即组内自动伸缩策略、组间资源回收策略、时间计划策略。

➤ 组内自动伸缩策略

组内自动伸缩是针对单独的应用而言的。

在运行过程中，组内自动伸缩策略会根据应用的当前负载动态来调整应用实际使用的资源，这里的动态调整主要分为伸和缩。

伸是指当一个应用资源负载较高时，系统可以自动地给这个应用添加虚

拟机，并且安装应用软件，以降低应用的整体资源负载，使应用能够健康地运行。

缩和伸是相对的，当应用资源负载很低时，系统可以自动减少应用使用的虚拟机，释放相应的资源，以达到应用间资源的有效复用和节能减排的目的。

➤ 组间资源回收策略

组间资源回收策略指的是，当系统资源不足的情况下，系统可以根据组间设置的资源复用策略，使优先级高的应用使用资源，使优先级低的应用释放资源。

➤ 时间计划策略

时间计划策略允许用户对不同的应用实现资源的分时复用。用户可以设置计划策略，使得不同的应用分时段地使用系统资源，比如白天让办公用户的虚拟机使用系统资源，到了晚间可以让一些公共的虚拟机占用资源。

虚拟机自动备份

虚拟机备份是使用备份软件，配合Hypervisor快照和CBT（Changed Block Tracking）功能实现的虚拟机数据备份方案。备份软件通过与Hypervisor配合，实现对指定虚拟机的备份。当虚拟机数据丢失或出现故障时，可通过备份的数据进行恢复。数据备份的目的端为本地虚拟磁盘或备份软件外接的共享网络存储设备（NAS）。

备份软件通过设置备份策略，实现虚拟机的自动定期备份。针对不同虚拟机或虚拟机组，可设置不同备份策略，最多支持200个备份策略：

- 支持对全备份、增量备份和差量备份分别设置不同的备份周期、备份时间窗口，如设置每周进行一次全备、每天进行一次增备，也可以只进行一次全备，后续一直进行增备；
- 设置备份数据保留时间，以自动清除过期备份数据；

- 设置备份策略优先级。

8.1.3 网络自动化

虚拟网络

虚拟化具备EVS功能，支持分布式虚拟交换，可以向虚拟机提供独立的网络平面。像物理交换机一样，不同的网络平面间通过VLAN进行隔离。这种技术具备如下特点：

- 同一宿主机上的不同虚拟机，如位于不同VLAN，则不能直接互通；
- 同一宿主机上的不同虚拟机，如位于相同VLAN，则可以直接二层互通，此时网络流量通过内存交换，不受任何网络带宽限制；
- 不同宿主机上的不同虚拟机，如位于相同VLAN，则可以通过外部交换机进行互通，就像没有虚拟化一样。

通过这种能力，您可以将VLAN视为独立的网络平面，通过向不同的虚拟机分配不同的VLAN，来实现各种业务间的隔离。

虚拟DHCP

虚拟DHCP就是一种DHCP服务，可以部署到任意的虚拟网络平面中。您可以透过此服务来管理虚拟机的IP地址分配。虚拟DHCP其实是一台独立的虚拟机，其上面运行的DHCP程序与物理DHCP服务并无功能上的显著区别。

由于虚拟DHCP服务是通过虚拟机部署的，所以可以随时依据需要灵活地部署到网络平面内。虚拟DHCP的部署过程也是自动化的，并不需要手动部署。

由于虚拟DHCP服务是服务于指定网络平面的，所以可以在不同的网络平面间使用不同的DHCP服务，这样就可以在不同的网络平面之间做到IP地址重叠。

在传统的DHCP服务上，虚拟DHCP服务追加了虚拟机DHCP数据的集中

管理能力，即虚拟机的MAC地址和IP地址绑定关系是集中存储在虚拟化的集中数据库中的。这样的好处就是，当某一个虚拟DHCP服务实例发生故障时，我们可以很方便地重新部署一个虚拟DHCP服务，然后把数据注入进去。这样使得虚拟服务的修复变得容易。

总结一下，虚拟DHCP服务具备如下优点：

- 按需随时部署回收；
- 支持IP地址重叠；
- 无状态，便于替换。

虚拟网关服务

使用虚拟三层网关服务，可以为子网提供三层路由的能力。三层网关会占用子网的网关地址，向子网内的虚拟机及其他设备提供三层路由的能力。从这一点讲，虚拟网关的能力与物理网络设备（交换机、路由器等）一致。

和虚拟DHCP服务一样，虚拟网关服务也是通过向虚拟网络平面部署虚拟机来实现虚拟网关服务的，即通过一台虚拟机为虚拟网络平面上的某个Subnet提供路由能力。

虚拟网关服务可以同时为多个子网提供网关能力。当虚拟网关服务同时为多个子网提供服务时，虚拟网关会自动配置这些子网间的默认路由。这样很方便，一个虚拟网关下的多个子网间默认就是可以路由的，通过虚拟路由功能控制网络间数据通信。

虚拟网关服务除了提供网关能力外，还提供NAT和NAPT的能力。基于此能力，其可以实现更加灵活的组网和业务。

虚拟网关的数据也是集中存储在虚拟化的数据库中的。虚拟网关的修复方式和DHCP服务一样，都是重新部署虚拟机，注入数据，这使得虚拟网关的修复变得很容易。

由于虚拟网关运行于虚拟机之上，其处理能力及带宽吞吐量会受宿主机上其他虚拟机的影响。对于网络压力不大的场景，可以使用虚拟网关来

应对，这样部署方便且成本低。对于网络压力比较大的场景，建议使用物理设备作为网关。

虚拟化一般支持将物理防火墙虚拟化为虚拟防火墙，同时提供虚拟网关的能力，推荐在性能和安全要求比较高的场景使用。

虚拟防火墙

云管理可以将物理的防火墙设备虚拟化为虚拟防火墙。虚拟防火墙可以分配给用户网络，放在网络的边界上，对外提供如下的功能：

- ACL隔离；
- 带宽限制；
- 三层网关；
- 默认路由；
- 弹性IP；
- SNAT穿越。

最终用户可以像配置物理防火墙一样方便地配置自己的虚拟防火墙。

由于虚拟防火墙使用的实际上是物理防火墙的硬件资源，所以其性能和可靠性是有保障的。

虚拟VPN

云管理支持利用虚拟防火墙的能力对外提供VPN能力。当前支持的VPN类型为：

- IPSec VPN Server to Server模式，用于跨广域的两个网络间的互通；
- L2TP over IPSec VPN，用于最终用户通过互联网访问私有网

络。

通过VPN能力，可以很方便地实现私有网络在云上的扩展，以及公网访问私有网络的能力。

由于虚拟VPN使用的实际上是物理防火墙的硬件资源，所以其性能和可靠性是有保障的。

虚拟负载均衡

虚拟负载均衡服务，即通过在虚拟机上部署软件的负载均衡器，向虚拟化基础设施提供负载均衡的能力。

由于虚拟负载均衡服务部署在虚拟机上，所以它可以很方便地随时部署销毁。同时，虚拟化资源池提供的虚拟负载均衡服务数据存储在虚拟化数据库中，所以在出现故障时可以很容易地恢复。

其支持将负载均衡器虚拟化来提供虚拟负载均衡，同时支持使用虚拟化提供的虚拟负载均衡能力来提供虚拟负载均衡服务。

负载均衡服务允许将来自公网的流量依据一定的负载均衡规则分发到多个业务处理虚拟机上。结合虚拟防火墙提供的弹性IP能力，带宽限制能力，可以很方便地控制Web应用的流量和压力。

虚拟化提供的虚拟负载均衡运行于虚拟机之上，其处理能力及带宽吞吐量会受宿主机上其他虚拟机的影响。对于网络压力不大的场景，可以使用虚拟负载均衡来应对，这样部署方便且成本低。

由于虚拟负载均衡运行于虚拟机之上，其处理能力及带宽吞吐量会受宿主机上其他虚拟机的影响。对于网络压力不大的场景，可以使用虚拟负载均衡来应对，这样部署方便且成本低。对于网络压力比较大的场景，建议使用物理负载均衡器。

云管理通过对负载均衡器虚拟化，提供基于硬件的虚拟负载均衡服务。由于使用独立硬件，性能要远远高于软件的虚拟负载均衡服务。

8.2 物理和虚拟化资源的统一管控

在云管理平台上，通过对各种物理资源、虚拟化资源数据统一建模，将资源以用户可见的资源池形式提供给上层应用，在接入不同的物理设备和虚拟化资源环境时，上层应用不感知（见图8-2）。

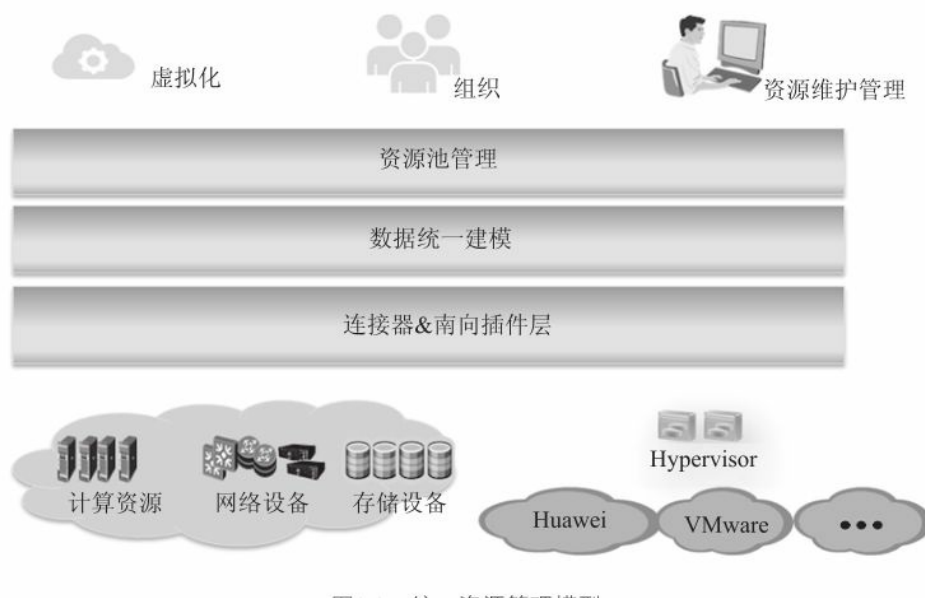


图8-2 统一资源管理模型

统一资源管理，支持发现其管辖范围内的物理设备（包括机框、服务器、存储设备、交换机、防火墙和负载均衡器等）以及它们的组网关系；支持将这些物理设备进行池化管理和集中管理，提供给上层应用管理使用，实现资源高效共享。

虚拟化资源管理可以统一管理不同厂商的虚拟化平台系统，如华为的FusionSphere、VMware的vSphere、微软的Hyper-V、思杰的Xen Server等虚拟化平台，提供不同资源的生命周期管理功能，包括虚拟机资源、虚拟网络资源、虚拟存储资源管理等。

通过资源池管理，其可提高基础设施资源的利用率和灵活性，提供统一的虚拟化资源管理能力，对上层应用发放屏蔽差异，实现虚拟资源集中管理，提升管理效率，降低运维成本。

采用南向插件机制，使云管理可以快速、便捷、可定制地实现不同硬件和虚拟化系统的对接。

8.2.1 物理资源管理

根据不同的资源对象，物理管理资源分为服务器资源管理、存储、网络设备。

服务器资源管理包括IBM、HP、ORACLE（Sun）、Fujitsu等支持基于Intel和AMD架构的x86主流服务器机。通过云管理平台，其对外提供统一的资源分配和管理。云管理平台支持多个虚拟化资源池的管理，一个资源池内的物理主机拥有同样类型的CPU架构或操作系统类型，以便进行业务弹性伸缩和动态调度。

网络资源基础管理提供对路由器、交换机、负载均衡器、防火墙、IP地址、虚拟网络设备等网络设备和资源的查询和配置管理，网络资源包括：资源编号，对应网络设备的基本信息，网络设备的管理和配置接口信息。将多个网络资源整合为一个整体，对外提供统一的网络资源分配和集中式管理。网络资源池应包含下面信息：网络资源池编号、网络资源类型（比如IP地址、交换机、路由器、负载均衡器、防火墙等）、网络资源池组成信息、网络资源池容量信息、资源池操作方式、资源池访问接口、系统域ID等。

典型的云管理平台，对网络资源能力包括：

- 支持对网络设备的自动发现；
- 支持设备拓扑图；
- 支持对网络设备（包括虚拟网络设备）的配置和管理，包括网络带宽容量、VLAN等资源的配置、查询、导出功能，支持通过全网资源统计，观测全网网络资源使用的状况；
- 支持网络资源（池）的容量管理，包括总容量、已使用的资源容量、可用资源容量等信息的管理和查询，对网络设备实时监测；
- 提供对网络资源（池）的生命周期管理（包括创建、修改、查询和删除）；
- 提供对网络资源（池）的操作和配置接口；

硬件设备接入协议一览表如表8-1所示。

表8-1 硬件设备接入协议一览表

分 类	设 备 类 型	接 入 协 议	告警上报协议
服务器	机框式服务器	SNMP、IPMI、SSH	SNMP
	裸金属机框式服务器	SNMP、IPMI	
	机架式服务器	IPMI、SSH	
	裸金属机架式服务器	IPMI	
存储	—	TLV、SMI-S	
交换机	—	SNMP、SSH	

➤ 服务器管理

通过服务器管理，可查看服务器名称和其他信息，管理IP地址、BIOS、CPU主频个数、内存大小、硬盘容量、网卡个数和网口个数等信息。

服务器操作管理对服务器进行安全下电、强制下电、安全重启、进入维护模式和退出维护模式等操作。

服务器监控管理监控服务器的如下指标：CPU占用率、内存占用率、网络流出、网络流入、磁盘I/O写入、磁盘I/O读出和服务器状态等，同时可以按周、月、年及自定义时段查询性能监控结果，也可以导出查询结果。

➤ 网络设备管理

查看交换机信息，如交换机的名称、管理IP地址、型号、类型和状态等信息。

查看交换机端口连接状态，如查看交换机每个端口的编号、状态、发送速率、接收速率、发送丢包率、接收丢包率、发送错误率和接收错误率等信息。

网络配置管理对系统网络进行配置和管理，如外部网络、组织网络和服务器BMC IP池等。

➤ 存储设备管理

查看存储设备的配置信息，如存储设备的名称、管理IP地址、型号、状态和磁盘数量等信息。

存储设备的监控管理，如查询存储设备的总容量和可用容量等。

[8.2.2 虚拟化资源管理](#)

云管理平台对虚拟化的管理包括虚拟化生命周期管理、虚拟网络管理、虚拟化存储管理和VPC业务。

虚拟化生命周期管理

➤ 创建虚拟机

管理员可以通过多种方式创建虚拟机，如通过应用创建虚拟机业务群、使用虚拟机模板来创建虚拟机（或批量创建多个虚拟机），通过自定义方式创建虚拟机，或者通过已有虚拟机克隆方式创建虚拟机。

➤ 销毁虚拟机

管理员可以通过删除应用来销毁虚拟机，将不再使用的虚拟机销毁，以释放系统资源。

➤ 虚拟机的维护

管理员可以对一个或多个虚拟机执行启动/唤醒、安全重启、强制重启、休眠、安全关闭和强制关闭等操作。

➤ 迁移虚拟机

管理员可以将虚拟机从一台主机上迁移到另一台主机上。

➤ 修复虚拟机

虚拟机操作系统异常后，管理员可以对虚拟机进行修复。修复虚拟机不会影响用户数据，确保用户信息不丢失。

➤ 创建虚拟机快照

虚拟机快照可保留虚拟机某一个时刻的状态，当虚拟机出现故障时，管理员可以使用快照将虚拟机恢复到创建快照的时刻点。

➤ 虚拟机资源调整

管理员可以根据业务负载调整资源的使用情况。虚拟机资源调整包括以下几点。

- 调整虚拟机的QoS：配置了虚拟机CPU QoS后，当虚拟机启动时，系统会根据当前资源使用情况为虚拟机的vCPU绑定物理CPU，直到虚拟机关机。
- 调整虚拟机CPU数目：管理员可以根据需要，增加或者减少虚拟机的CPU核数，以便满足虚拟机上业务负载发生变化时，对计算资源的不同需求。
- 调整内存大小：管理员可以根据需要增加或者减少虚拟机的内存容量，以便满足虚拟机上业务负载发生变化时对内存的需求。
- 增加或修改虚拟磁盘：管理员通过增加虚拟磁盘或修改磁盘容量，满足业务对虚拟机磁盘容量变化的需求，实现存储资源的灵活使用。
- 删除虚拟磁盘：管理员通过删除虚拟磁盘，释放不使用的磁盘空间，满足业务对虚拟机磁盘容量变化的需求，实现存储资源的灵活使用。
- 增加或修改网卡：管理员通过增加虚拟网卡或修改网卡属性，调整虚拟机的网络属性，实现网络资源的灵活使用。
- 删除网卡：管理员通过删除虚拟网卡，释放不使用的网卡，实现网络资源的灵活使用。
- 增加USB控制器：管理员可以为虚拟化环境中的虚拟机添加USB控制器，添加USB控制器后，虚拟机就可以绑定USB设备，实现与个人电脑同样的操作。
- 删除USB控制器：管理员可以删除虚拟化环境中的USB控制器，释放虚拟机所占用的USB控制器资源。
- 绑定或解绑定USB设备：管理员将虚拟机与主机上的USB设备进行绑定，使得虚拟机能够访问USB设备，满足用户需求；也可以解绑定，及时释放虚拟机占用的USB设备资源。
- 调整虚拟机磁盘的I/O上限：管理员可设置虚拟机每个磁盘的I/O上

限，以避免某个虚拟机的磁盘I/O过大，影响其他虚拟机的性能。

- 虚拟机性能监控：通过云平台系统，可以获取虚拟机CPU占用率、内存占用率、网络流速和磁盘I/O等信息，还可以按周、月、年及自定义时段查询性能监控结果。

虚拟网络管理

虚拟网络管理包括子网管理、VLAN池管理、VPC管理和虚拟防火墙。云平台的子网管理，支持子网下虚拟机的二层隔离。当组网模式采用二层、三层时均可根据用户需要配置VLAN池，在组网模式为三层组网时添加的VLAN池只用于隔离二层网络。VPC为应用发放提供了一个独占并且完全隔离的网络容器，可以在VPC内添加虚拟防火墙和各种类型的网络。

软件的虚拟防火墙是指通过VSA虚拟机提供防火墙功能，该防火墙能为VPC提供DNAT和VxLAN业务，也能够提供类似于弹性IP的功能，只不过“弹性IP”是直接从外部网络，而不是公网IP池中获取的。硬件虚拟防火墙是在物理防火墙设备上创建的虚拟防火墙，对防火墙性能要求较高时可以选择此类型，并且能够为VPC提供弹性IP、DNAT、ACL和VPN功能。

VPC能够为组织提供安全、隔离的网络环境，可以在VPC中定义与传统网络无差别的虚拟网络，以满足业务部署要求。在VPC中，创建的虚拟机或应用被部署到网络当中，可用于提供服务。

VPC提供三种网络用于部署业务资源。

➤ 直连网络

直连网络与外部网络相连，其自身不包含任何网络资源，在直连网络中创建虚拟机或应用时实际使用的是外部网络中的资源。外部网络可以是公司现有网络或者公网，当外部网络为公司现有网络时，云管理与公司现有网络对接，虚拟机可分配到公司现有网络的IP地址资源。当外部网络为公网时，其直连网络中的虚拟机具有直接访问公网的能力。

➤ 内部网络

由于内部网络和其他的网络是隔离的，因此内部网络中可以部署对安全

性要求较高的业务，例如，部署一个三层网站时，可将数据库所在服务器部署在内部网络中。

➤ 路由网络

路由网络具有灵活的互通能力和多种业务功能，基于虚拟防火墙的路由网络能够与VPC中的其他路由网络互通，或者绑定弹性IP与公网进行通信。除了弹性IP，路由网络还能提供ACL、DNAT和VPN业务，以满足业务部署要求。在创建路由网络前，需要先为VPC申请虚拟防火墙（见图8-3）。

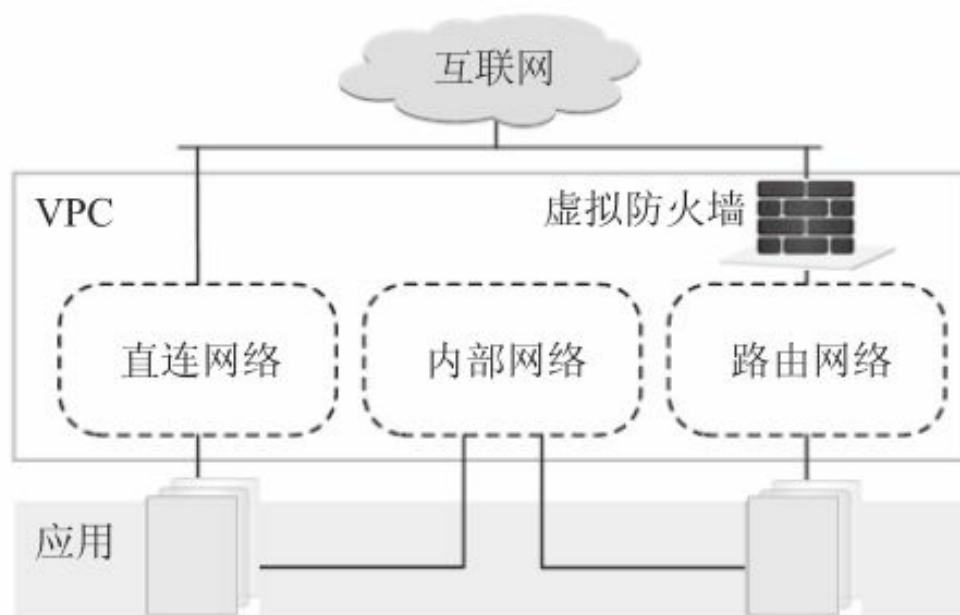


图8-3 VPC网络结构

VPC的业务包括以下几点。

➤ 弹性IP

弹性IP地址是一个公网IP地址，该IP地址可以与VPC内任何一个路由网络中的内部IP地址绑定。这个内部地址可以是虚拟机的IP地址、VLB的虚拟IP地址或浮动IP地址。例如，为VPC内的Web服务器绑定弹性IP后，公网用户通过访问弹性IP地址使用Web服务，如图8-4所示。

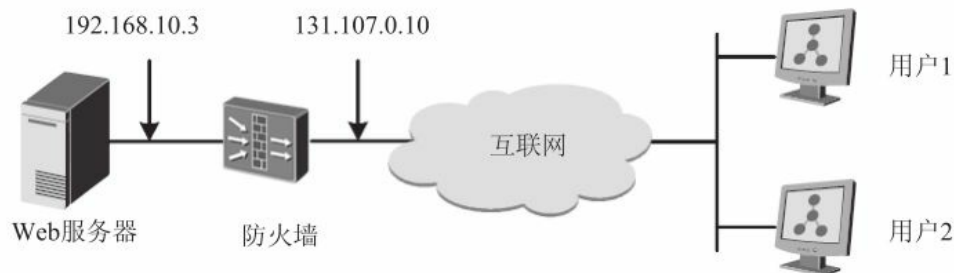


图8-4 弹性IP

➤ DNAT

当VPC内部需要提供对外服务时，公网用户发起连接请求，由防火墙上的网关接收这个连接，然后将连接转换到内部，此过程是由带有公网IP的网关替代内部服务来接收外部的连接，然后在内部做地址转换，此转换称为DNAT，主要用于内部服务对外发布（见图8-5）。

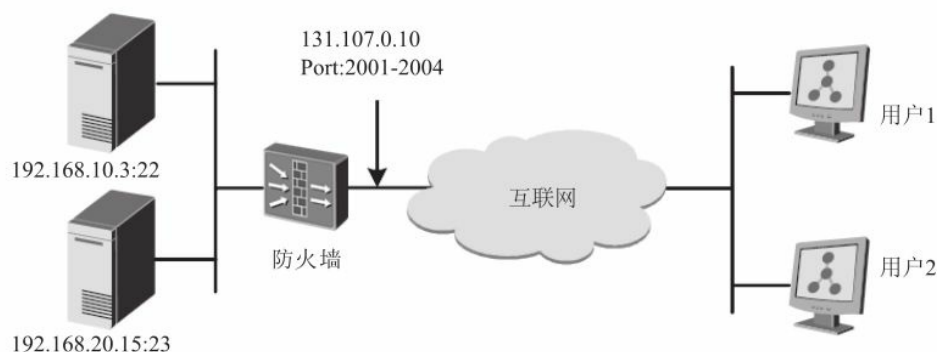


图8-5 DNAT

➤ SNAT

内部地址单向发起请求访问公网上的服务时（如Web访问），内部地址会主动发起连接，由防火墙上的网关对内部地址做地址转换，将内部IP地址转换为公网IP地址。这个由网关完成的地址转换称为SNAT，主要用于内部共享IP访问外部，如图8-6所示。

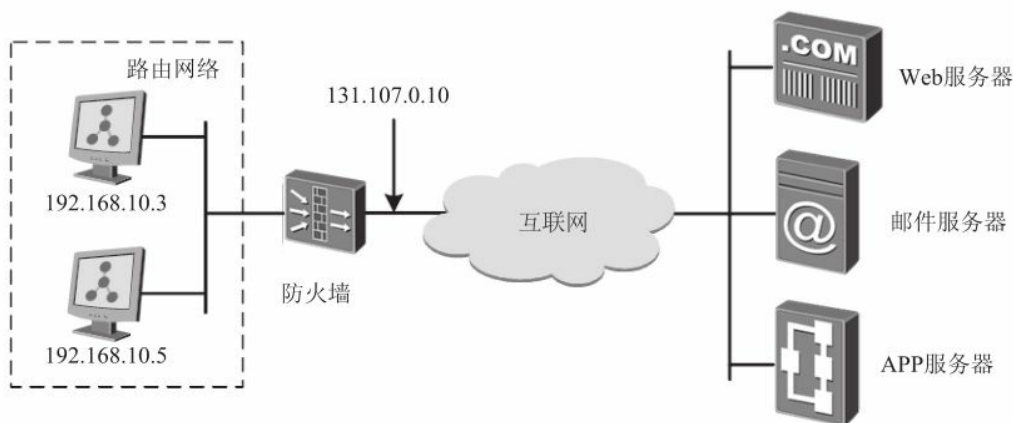


图8-6 SNAT

➤ VxLAN

VPN业务用于在公网之上建立一条安全、稳定的通信隧道，将分布于不同地域的企业或个人连接起来，并保证通信隧道内发送和接收数据的安全性。云管理需支持IPsec和L2TP两种类型的VPN，将处于公网中的用户接入到VPC的路由网络，使用户与路由网络中的服务器互通。

➤ 安全组

安全组用来实现安全组内和组间虚拟机的访问控制，加强虚拟机的安全保护。安全组创建后，管理员可以在安全组中定义各种访问规则，当虚拟机加入该安全组后，即受到这些访问规则的保护。

例如，在同一VLAN下的两个部门之间相互隔离，同一部门之间的虚拟机可以相互访问，但是所有虚拟机都可以与服务器通信。解决方法如图8-7所示，分别为部门1、部门2创建安全组1、安全组2，且安全组为组内互通；为安全组1和安全组2添加安全组规则，允许服务器的IP地址段访问安全组。

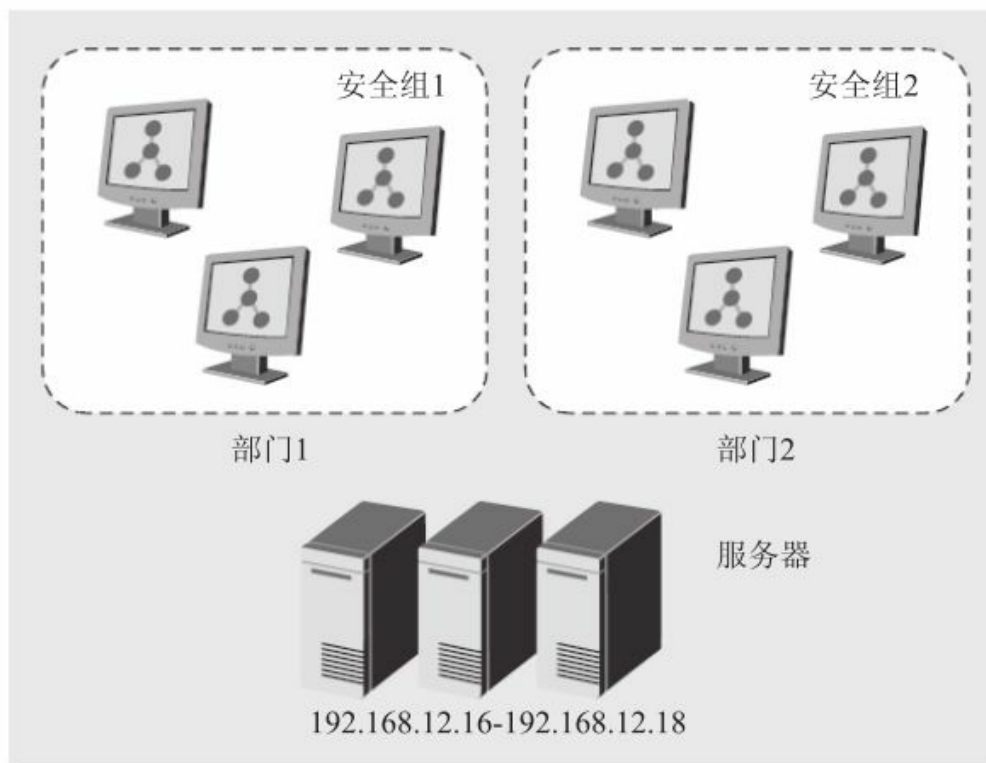


图8-7 虚拟机隔离

虚拟存储管理

虚拟化存储将不同厂商的各种物理存储设备连成一个逻辑存储池，对外提供业务并存储空间管理，根据存储空间类型不同，可分为文件资源、块数据资源和对象资源。所有的存储设备将被统一管理，提供数据复制、快照、克隆、远程灾备等业务。

云管理平台提供数据存储空间（包括文件、块和对象）的生命周期管理接口，对存储设备进行信息记录和综合管理，包括存储空间的创建、删除、加载、卸载、查询等；支持各种主流品牌存储设备的接入和管理，采用行业标准的开放接口，将不同厂家、不同型号的存储设备接入到统一存储虚拟化层，存储虚拟化层对这些存储设备进行发现、管理、分配，存储虚拟化后，对上层应用提供统一的存储访问格式。

云管平台支持跨地域的存储池的统一监控管理，实现远程管理能力。

8.2.3 资源集群管理

集群是由多个计算服务器组成的，集群内的服务器作为一个整体来工作和进行管理。一个虚拟化环境中可以有多个集群。虚拟机在创建时被指定在某一个集群内，根据一定的原则（如负荷分担）选择一个主机，在此主机上创建虚拟机。对于集群内的计算资源共享，虚拟机可以从一个物理主机迁移到另一个物理主机。资源集群管理的功能包括创建资源集群、删除资源集群、扩容资源集群、减容资源集群、监控资源集群性能。

➤ 创建资源集群

选择计算资源、存储资源、网络资源来创建资源集群，在创建过程中可以进行高级配置，配置资源集群的HA和主机内存复用信息。

➤ 删除资源集群

删除资源集群后，相应的计算资源、存储资源、网络资源会被释放。

➤ 扩容资源集群

给已经发放的资源集群添加服务器、数据存储资源。

➤ 减容资源集群

对已经发放的资源集群，可以删除其中的服务器、网络和数据存储来释放资源。

➤ 查看资源集群

显示资源集群的基本信息包括集群名称，所属的域、虚拟化环境及资源分区，集群中主机总数，故障主机数，虚拟机总数，故障虚拟机数等。

➤ 资源集群性能监控

可以按周、月、年、自定义时间段查询资源集群性能监控，监控指标有平均CPU趋势对比、平均内存趋势对比、平均网络流量趋势对比（网络流出）、平均网络流量趋势对比（网络流入）。

➤ 集群HA（High Available）

虚拟机支持热迁移，支持在一个计算集群内自由迁移虚拟机。在虚拟机迁移期间，用户业务不会有任何中断。如果迁移失败，目的端的虚拟机将被销毁，而用户仍可以使用源端虚拟机。该功能可避免因服务器维护造成的业务中断，降低数据中心的电能消耗。

虚拟机支持故障迁移，该功能支持虚拟机故障后自动重启。用户创建虚拟机时，可以选择是否支持故障重启，即是否支持HA功能。系统周期检测虚拟机状态，当物理服务器故障引起虚拟机故障时，系统会将虚拟机迁移到其他物理服务器重新启动，保证虚拟机能够快速恢复。重新启动的虚拟机，会像物理机一样重新开始引导，加载操作系统，所以发生故障时未保存的内容将丢失。

8.2.4 虚拟机资源管理

虚拟机资源管理包括对虚拟机生命周期管理和虚拟化格式转换。

➤ 虚拟机生命周期管理

虚拟机创建，即接收业务运营系统的申请虚拟机请求，检查资源池及分区资源是否满足申请，根据运营参数在对应的存储池分配相应的虚拟机资源。其支持通过模板快速批量创建虚拟机，或者从已有虚拟机克隆虚拟机。

➤ 虚拟化格式转换

P2V（Physical Machine to Virtual Machine）是指将物理服务器连同上面安装的操作系统和应用转换为虚拟机，常用于将已有应用系统由物理服务器平台迁移到虚拟机平台。其支持离线或者在线地将物理服务器上的操作系统（Windows/Linux）迁移到虚拟环境。

P2V转换支持离线和在线两种方式。在线方式是物理服务器上的应用系统在对外提供服务的过程中实现P2V转换，转换完成后切换到虚拟机继续提供服务，在线P2V转换应保证业务不中断。离线方式是在转换过程中，应用系统不对外提供服务。

V2V是指将虚拟机从一种虚拟机格式转换成另一种虚拟机格式，支持跨

主流x86虚拟化平台的虚拟机间的转换，可以自动检测并替换原有的虚拟硬件设备以保证虚拟机迁移后的正常运行。转换成功后，虚拟机支持正常运行，数据不丢失。

8.3 基于网络的硬件即插即用的自动化机制

8.3.1 设备自动发现和部署

目前IT业务普遍采用数据中心集中管理，统一对外提供软硬件资源，整个云数据中心的硬件快速扩容，更换硬件将直接影响这个数据中心的业务的故障恢复和系统扩容的效率。因此，系统基于网络的硬件即插即用在云基础设施维护中变得至关重要。

系统从硬件插入机框上电到提供计算服务资源，全流程自动化完成。

即插即用的自动化机制包括：

- 硬件的自动发现；
- 基础管理网络的配置（服务器端口网络属性的配置、物理交换机网络配置和IP地址自动分配）；
- 操作系统的自动安装，系统可根据用户配置，自动完成各种操作系统的安装；
- 硬件监控信息可自动检测并上报，监控指标包括CPU、内存、磁盘I/O、网络带宽和物理设备的静态硬件信息。

即插即用方便系统的扩容，减轻人为操作带来的误操作风险，并大大减少系统扩容的时间，满足业务的快速扩容上线。

服务器的自动化发现流程如图8-8所示。

本地资源管理由物理机管理、网络管理、模板/镜像管理、PXE Server、PXE Client共同完成物理机的自动部署。

- 物理机管理：负责物理机的发现，通过IPMI协议设置BOIS启动方式、控制上/下电；按物理机硬件配置进行池化管理，根据租户请求选择合适的物理机进行发放。
- 网络管理：自动配置物理机所在的接入交换机和汇聚交换机，使物理机接入租户的指定网络，与其他物理机或虚拟机互通。
- DHCP Server：自动分配物理机IP地址，以实现物理机与本地资源管理通信。
- PXE Server/PXE Client：通过PXE协议控制物理机自动安装租户指定操作系统。
- TFTP Server：提供SimpleOS下载服务。

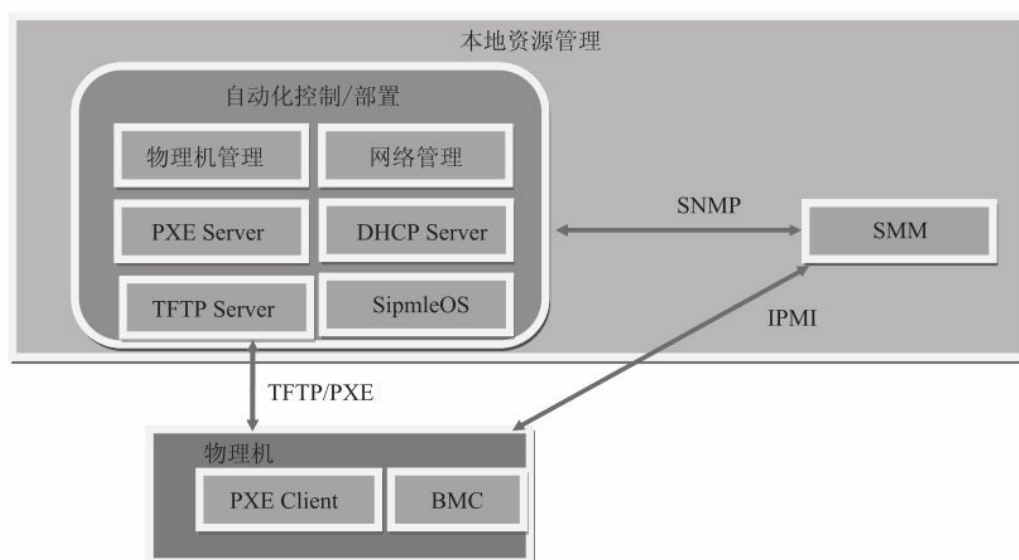


图8-8 服务器的自动化发现流程

- SMM：机框管理单元。
- BMC：服务器管理单元。

其具体流程如下。

- 步骤1：服务器接入系统后，系统通过机框管理单元以网络启动，并重启服务器。
- 步骤2：服务器重启后，发送DHCP请求，DHCP服务器在收到DHCP请求后，发送给物理机分配IP地址的DHCP回应包，内容包括物理机的IP地址、TFTP服务器的IP地址和开机启动文件。
- 步骤3：服务器通过TFTP通信协议从系统下载开机启动文件和定制的简单操作系统，并在内存中完成该定制操作系统的安装和启动。
- 步骤4：通过新安装的定制操作系统获取服务器静态信息（包括CPU、内存规格、版本和网络地址信息）。
- 步骤5：将新发现服务器的监控信息上报给系统呈现，并将服务器状态标记为就绪状态，方便用户查找和申请物理机服务。

服务器自动化部署如图8-9所示。

本地资源管理由物理机管理、网络管理、模板/镜像管理、PXE Server、PXE Client共同完成物理机的自动部署。

- 物理机管理：通过IPMI协议设置BOIS启动方式、控制上/下电；按物理机硬件配置进行池化管理，根据租户请求选择合适的物理机进行发放。
- 网络管理：自动配置物理机所在的接入交换机和汇聚交换机，使物理机接入租户指定网络，与其他物理机或虚拟机互通。
- DHCP Server：自动分配物理机IP地址，以实现物理机与本地资源管理通信。
- PXE Server/PXE Client：通过PXE协议控制物理机自动安装租户指定操作系统。

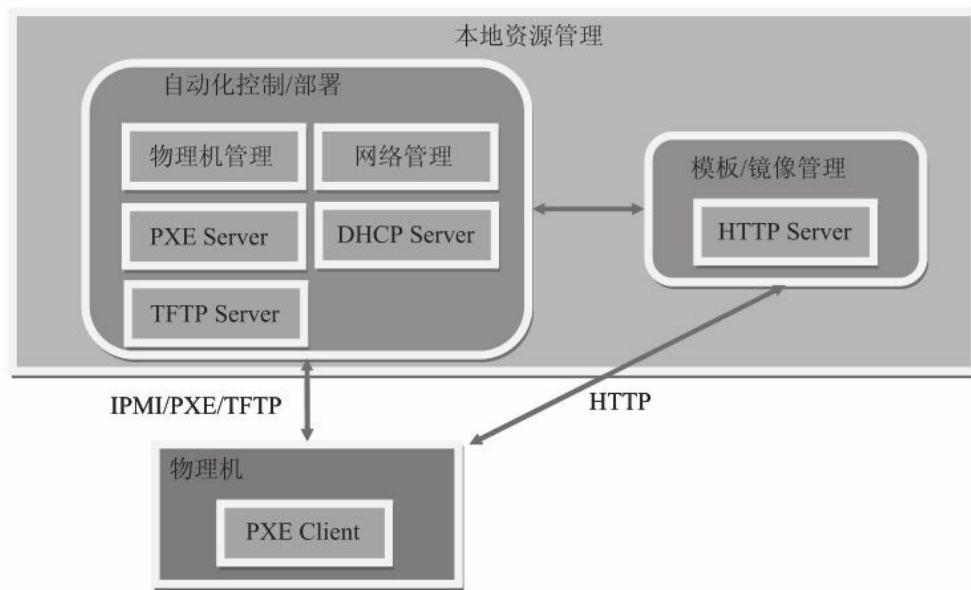


图8-9 服务器自动化部署

- **TFTP Server:** 提供引导程序下载服务。
- **HTTP Server:** 提供操作系统镜像下载服务。

其具体流程如下。

- **步骤1:** 用户申请物理机后，资源池管理通过BMC口设置从网络启动，并重启物理机。
- **步骤2:** 物理机重启后，发送DHCP请求，DHCP服务器在收到DHCP请求后，发送给物理机分配IP地址的DHCP回应包，内容包括物理机的IP地址、TFTP服务器的IP地址和开机启动文件。
- **步骤3:** 物理机通过TFTP通信协议从服务器下载开机启动文件。启动文件接收完成后，将控制权转交给启动块，完成PXE启动。
- **步骤4:** 启动块从镜像管理下载操作系统镜像到物理机，启动操作系统镜像，自动完成操作系统安装。

8.3.2 服务器自动化

资源池管理模块负责处理用户的服务部署要求，实现部署和配置自动化。可根据服务对应的资源容量、规格、QoS要求，自动完成x86服务器、小型机、存储设备、网络设备、虚拟机等的自动部署和配置，也能够完成数据库、中间件、应用软件、系统补丁等自动化安装、配置。

从裸机到应用的全流程的自动化部署支持如下服务。

- 操作系统部署：支持Windows、Redhat、Suse、Ubuntu等操作系统自动部署。
- 软件安装：支持数据库、Web中间件、JDK、应用软件等软件自动安装，可以在物理机申请时和申请后安装。
- 更新操作系统补丁：物理机部署操作系统后，操作系统如果有新的补丁，用户可以选择自动化更新补丁。

操作系统自动化

服务器操作系统自动部署通过PXE机制实现，完成IP分配、操作系统安装、操作系统补丁安装，总体方案如图8-10所示。

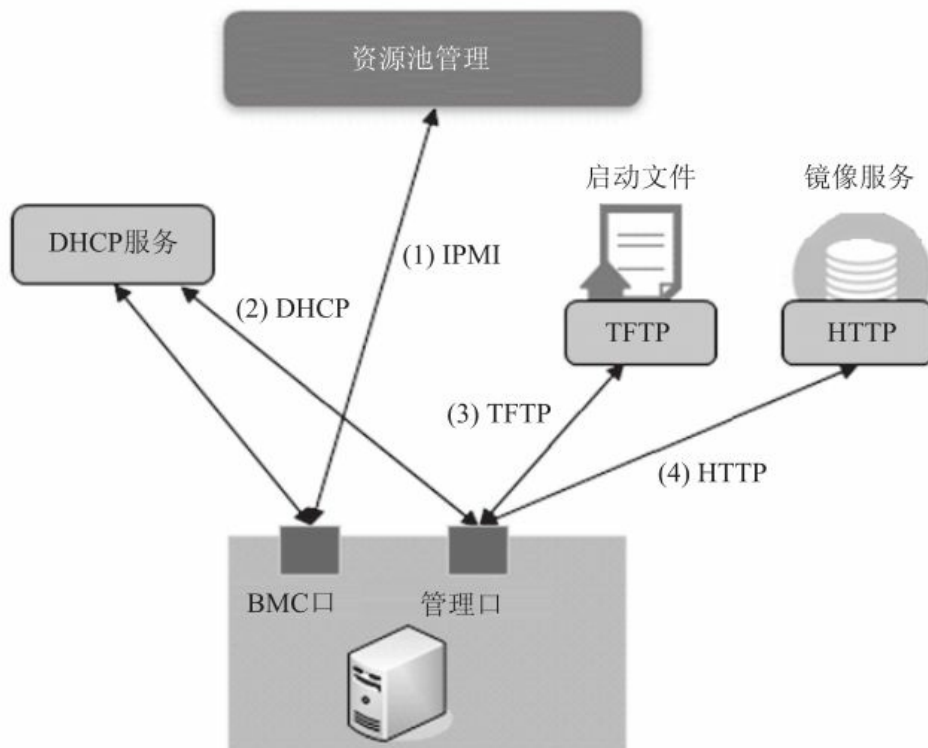


图8-10 总体方案

其具体流程如下。

- 步骤1：用户申请物理机后，资源池管理通过BMC口设置从网络启动，并重启物理机。
- 步骤2：物理机重启后，发送DHCP请求，DHCP服务器在收到DHCP请求后，发送给物理机分配IP地址的DHCP回应包，内容包括物理机的IP地址、TFTP服务器的IP地址和开机启动文件。
- 步骤3：物理机通过TFTP通信协议从服务器下载开机启动文件。启动文件接收完成后，将控制权转交给启动块，完成PXE启动。
- 步骤4：启动块从镜像管理下载操作系统镜像到物理机，启动操作系统镜像，自动完成操作系统安装。

软件安装/更新

数据库、中间件、应用软件等安装/更新部署的自动化方案如图8-11所示。

自动化具体流程如下。

➤ 步骤1：物理机部署成功后，物理机上的软件安装代理（注：制作物理机镜像时，已经将软件安装代理包含在物理机镜像中）自动向资源池管理发起注册。

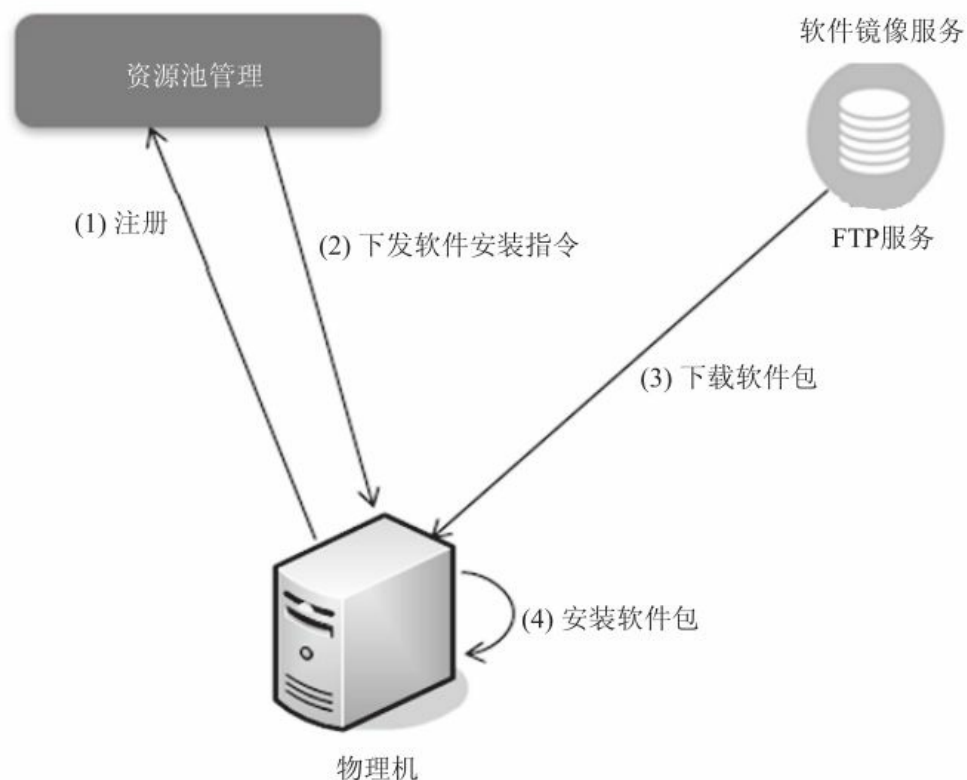


图8-11 数据库、中间件、应用软件等安装/更新部署的自动化方案

➤ 步骤2：资源池管理向软件安装代理推送软件安装指令。软件安装指令中描述了从哪里获取软件包、软件包安装命令和安装参数。

➤ 步骤3：软件安装代理连接软件镜像服务器，通过FTP下载软件包。为了保证软件包下载的可靠性，支持下载断点续传，失败重试。

- 步骤4：通过代理执行软件安装/命令进行软件安装。

8.4 异构硬件的统一接入管理

云管理的硬件异构包括服务器异构、存储设备异构和网络设备异构。不同厂商的设备接口存在较大的差异，那么怎样快速、简单地接入不同设备是资源管理优先考虑的问题（见图8-12）。

异构硬件通过插件技术，可以在不需要升级版本的情况下，通过适配方式支持，业务可不受影响。硬件资源管理系统从逻辑上分为三层，分别为O&M应用服务层、设备适配层和设备通信层。O&M应用服务层对外提供硬件资源操作管理维护的接口；设备适配层负责封装对硬件资源的监控和业务配置操作，对上提供统一的服务接口，屏蔽与异构设备的通信协议差异；设备通信层负责实现与设备的通信，实现设备信息的查询和配置命令的下发。对外提供Web形式的操作维护平台，用户可通过Web浏览器进行远程管理，同时对外提供开放统一的REST北向接口，提供硬件资源监控和业务配置接口。

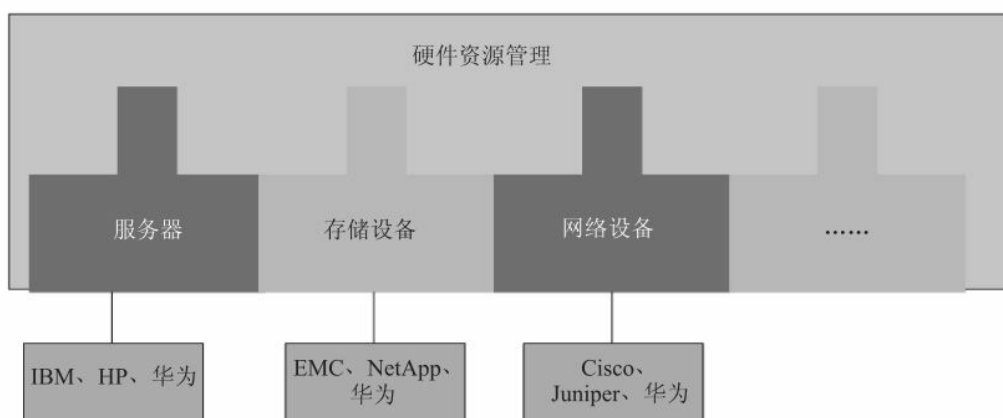


图8-12 硬件资源管理

服务器接口协议支持IPMI、SSH、SNMP，存储设备接口协议支持TLV、SMI-S，网络设备接口协议支持SSH、SNMP。

服务器的配置信息包括名称、位置、管理IP地址、BIOS、CPU主频、个数、内存大小、硬盘容量、网卡个数、网口个数。

交换机的配置信息包括位置、交换机管理IP地址、型号、类型、状态。

存储设备的配置信息包括存储设备位置、产品型号、状态、管理IP地址、磁盘数量。

服务器设备的维护包括上电，下电，安全重启，安全下电，强制下电，进入维护模式，退出维护模式，一键式上电、下电所有服务器。

对于异构的物理设备，系统提供强大完备的监控能力，可以呈现设备的实时监控指标，也可以呈现设备的历史监控指标，如监控服务器的CPU、内存和磁盘占用率、网络设备的流量、存储设备的IOPS等。

异构物理设备的故障告警支持对各类设备的告警进行统一呈现和统一操作，如磁盘、风扇故障告警、温度过高告警、设备离线告警等。

8.5 服务目录和应用管理

服务目录管理是指根据目录类别对资源服务进行分级管理和显示。根据应用场景，服务目录分为以下两部分。

- 服务申请：给管理员提供用户自助管理界面，管理员可以一键式快速创建应用，还可以方便查看应用的部署报告和创建进度。
- 应用管理：支持服务目录的生成、发布、修改、删除、查询和导入导出，支持分级目录管理；监控应用日志，监控应用运行情况和变更，便于管理员及时发现和定位应用故障。

8.5.1 应用发布流程介绍

服务目录为用户提供了方便的、获取资源的途径，用户可以通过服务目录自动化获取资源，并在资源上部署用户需要的应用。

业务管理员在得到系统管理员授权后，可以自助进行服务发放和管理。业务管理员可以自定义服务模板，灵活部署服务，并且可以共享模板。

云管理平台提供用户自助门户访问服务模板目录，可以根据服务模板创建应用、管理发放应用的功能（见图8-13）。

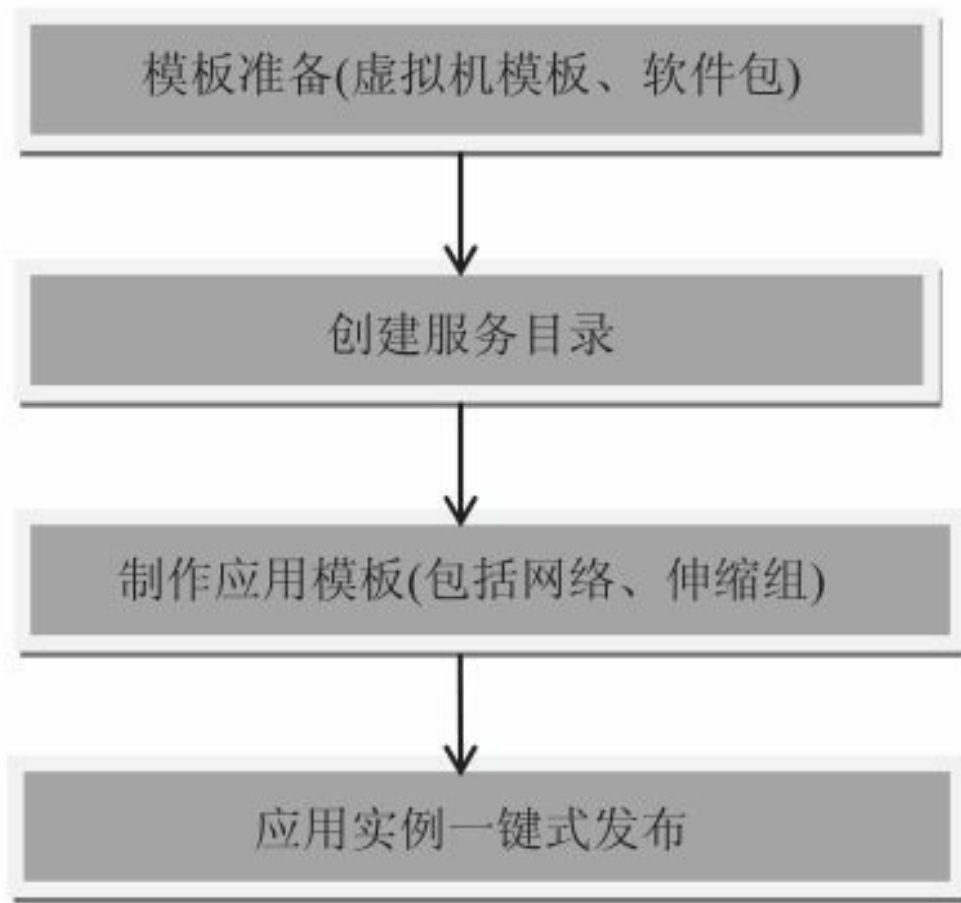


图8-13 应用发布流程

➤ 模板准备

管理员可以根据企业不同的应用需求，上传并管理应用的软件包，为应用服务模板的设计提供应用软件组件。

➤ 创建服务目录

管理平台通过服务目录来管理不同的应用模板，通过服务目录做分类处理，便于后续的统一维护和管理。

➤ 制作应用模板

对于应用模板的制作，业界目前有两种方式：伪编程模式和拖拽交互式。伪编程模式，通过管理平台提供的大量接口，可以灵活地实现业务逻辑和应用关系，但是对管理员要求较高，需要经过专业的理论和实践

学习；拖拽交互式，将应用制作过程模块化、简单化，通过工具在画布上拖拽对应的设备，实现应用模板的制作和调试过程，对管理员要求低，可以快速入手。

随着云计算走进中小企业，拖拽式交互是云管理应用部署的发展方向，所以本文通过这种方式来说明应用的部署。

应用部署模板设计工具提供用户可视化设计，用户可以简单地通过在画布上拖拽的方式，方便快捷地完成应用部署模板的设计（见图8-14）。同时通过资源间的连线，用户可以方便地定义应用中的资源依赖关系，然后用户就可以通过设计好的模板发放应用了。

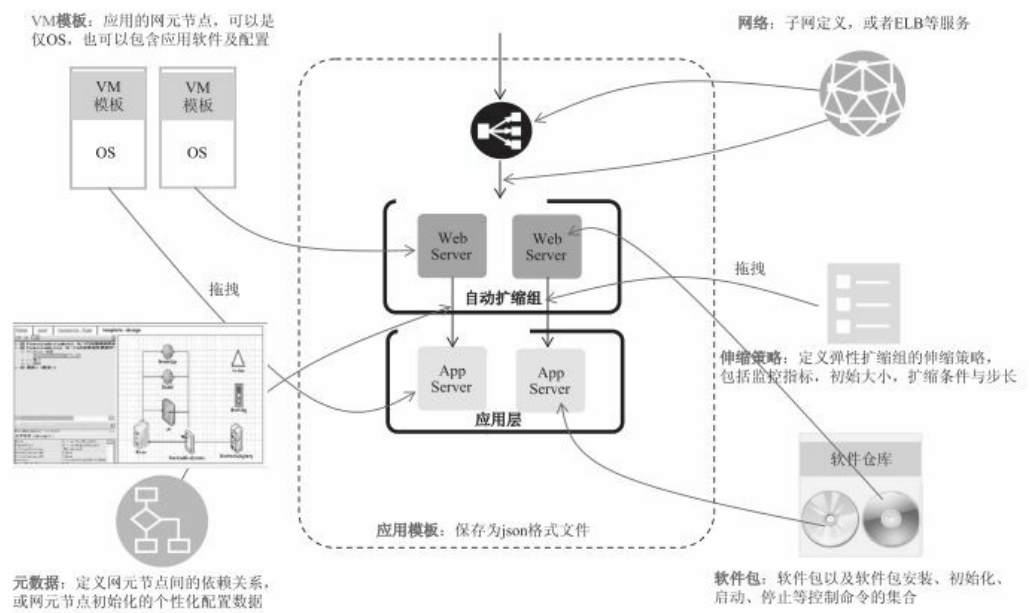


图8-14 应用部署模板设计工具

服务模板由虚拟机模板、软件包、网络和伸缩组等构成，虚拟机模板中可以定义初始化命令、启动命令和停止命令。软件包中可以定义安装命令、卸载命令、启动命令和停止命令。伸缩组中可以定义伸缩策略。

➤ 应用实例发布

其提供用户自助门户访问服务模板目录，一键式快速创建应用以及查看应用的部署报告和创建进度。

业务管理员通过自助门户访问服务模板目录，根据应用需求选择相应的

服务模板，配置应用网络以及选择应用的管理员，快速部署应用。在应用部署过程中，业务管理员可以查看应用部署报告和实时进度。

在应用部署中，VM的创建和软件分发流程如图8-15所示。

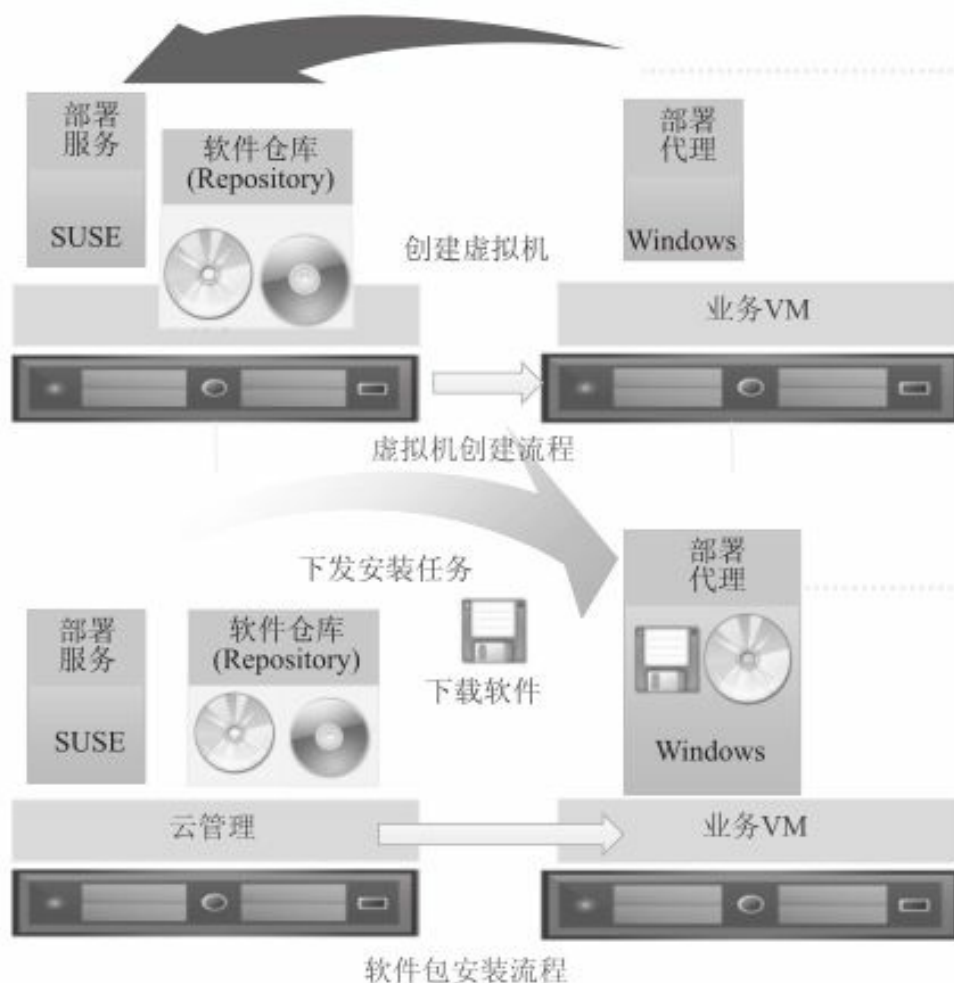


图8-15 VM的创建和软件分发流程

应用软件自动部署原理

- 部署服务根据应用模板中定义的虚拟机和网络关系以及虚拟机模板创建业务虚拟机并分配网络，业务虚拟机创建完成后，操作系统即安装完成。
- 业务虚拟机上已经安装了部署代理（部署代理在制作虚拟机模板的过程中被制作到模板当中，因此虚拟机创建出来后部署代理已

经在运行)。部署代理启动后，会与部署服务通信，部署服务将安装任务推送给部署代理。

- 部署代理根据安装任务，去软件仓库（Repository）中下载对应的软件，然后进行安装。

8.5.2 应用管理原理

应用管理提供用户管理发放的应用的功能，包括管理应用的生命周期、查看应用的监控日志以及通过应用拓扑实时查看和管理应用内部虚拟机、网络等。

应用管理包括：

- 管理应用生命周期，提供管理员对应用进行启用、挂起、修改和删除；
- 应用监控日志，提供管理员监管应用运行情况和变更操作的功能，便于管理员及时发现和定位应用故障；
- 应用拓扑管理，管理员可以可视化地查看应用内部的组网结构，管理应用虚拟机。

在企业实际使用中，应用会周期出现业务量的上升和下降，比如第三方支付应用，在圣诞、新年等重大节日前，业务量会比平时增加几倍。所以，应用发布后，怎样支持业务的弹性伸缩，是企业IT运维部门考虑的重点。

从企业应用的场景看，弹性伸缩可以分为三类。

- 组内自动伸缩策略：针对单独的应用而言，可根据应用当前的负载动态调整应用实际使用的资源，当应用资源负载较高时，自动添加虚拟机并且安装应用软件；当应用的资源负载很低时，自动释放相应的资源。
- 组间资源回收策略：当系统资源不足的情况下，系统可以根据组间设置的资源复用策略，使优先级高的应用使用资源，使优先级

低的应用释放资源。

➤ 时间计划策略：时间计划策略允许用户对于不同的应用实现资源的分时复用。用户可以设置计划策略，使得不同的应用分时段地使用系统资源，比如白天让办公用户的虚拟机使用系统资源，到了晚间可以让一些公共的虚拟机占用资源。

8.6 面向云管理的ITSM

服务运维模块提供统一的服务管理流程和支撑功能，实现运维管理人员对日常服务运维的集中管理。系统将服务运维管理数据集中整合，并以适当的形式呈现，支持维护人员进行运维故障定位、诊断和解决等运维活动，为运维管理提供基本信息。服务运维模块的建设目标包括以下几点。

➤ 以ITIL V3为基础开展服务运维管理，推广IT服务管理理念。从管理要求和技术手段两个层面提高信息化的工作效率和服务水平，从而满足日益增长的信息技术和服务的需求，提高响应效率，促进用户满意度提升。

➤ 通过优化运维管理体系，明确IT服务管理体系组织结构和相应的岗位和职责，对IT系统及相关资源进行集中、统一、真实、有效的管理。

➤ 对现有的IT运维体系进行梳理并实施规范化的流程设计。执行完善的事件、服务请求、变更、配置、信息发布管理流程。通过上述流程的建设和推广，后续可以逐步将IT服务运维管理不断深化，扩大服务管理流程的实践范围。

➤ 对运维工作进行量化，建立配套的衡量指标和日常管理制度，实现有效的目标考核机制，为合理配置资源提供依据，缩减运行成本。

➤ 提供集中整合的服务运维管理数据，为管理决策提供支持。

服务运维模块实现的服务管理流程包括服务请求管理、事件管理、变更

管理、配置管理、信息发布管理，各管理流程之间紧密关联，并与其他模块及子系统进行有效衔接。

➤ 与服务运营模块集成：服务运营模块面向用户提供自助服务，当用户提交的服务请求、资源变更涉及人工执行环节，或提交投诉建议、故障申告时，将通过系统间接口发送至服务运维的相关管理流程，自动创建工单，并将处理状态、处理结果等返回服务运营用户界面，从而形成端到端的闭环服务管理流程。

➤ 与全局资源管理模块集成：服务运营模块采用定时同步机制，保证配置管理信息与统一CMDB的数据一致性。

➤ 本地资源管理系统集成：本地资源管系统的告警会统一汇聚到全局的运维中心，运维人员在全局的运维中心内集中监控资源的状态，并处理各个数据中心汇聚上来的告警。

根据预定义事件单生成规则，全局的运维中心中的故障告警会在服务运营管理器中触发告警类事件单的自动创建，由运维人员按照预定义的事件处理流程，进行事件的处理。

事件处理流程结束时，服务运营管理器可以按照预定义规则，触发本地资源管理系统服务自动化部件执行事件的自动恢复。

8.7 云平台第三方App资源使用计量

资源使用计量是指资源池管理平台从资源池系统或用户订单获取用户资源使用信息，并根据用户使用资源信息生成资源使用计量文件。资源使用计量可以按照资源类别、业务系统和用户总拥有资源进行汇总统计。计量文件可以给统计分析等模块提供信息，用于用户资源使用信息的展示、统计及分析等。

计量话单包括计次话单、审计话单、负载均衡流量话单和弹性IP流量话单。

➤ 计次话单：在用户进行虚拟机、块存储、备份和各类网络资源的申请、修改、释放、冻结和解冻等操作时，记录相关信息到文

件，然后发送给话单服务器，生成用户话单。其包括的信息有：话单长度、流水号、话单类型、服务类型、资源标识、操作标识、操作时间、资源信息等。

➤ 审计话单：是对计量话单的补充，在计次信息遗漏时，为计费提供依据。这些话单可以协助计费中心实现按时段收费、对账等，审计对象包括实例运行时间、存储使用时间、弹性IP占用时间等，其包括的信息有话单长度、流水号、话单类型、服务类型、资源标识、操作标识、操作时间、审计信息等。

➤ 负载均衡流量话单：系统定时记录每个负载均衡器的流量信息到文件，然后发送给话单服务器，用于生成用户话单。其包括的信息有：创建时间、报文个数、报文信息列表（含流量字节数）等。

➤ 弹性IP流量话单：系统实时收集来自网络的弹性IP流量信息，记录到文件，然后发送给话单服务器，生成用户话单。其包括的信息有：创建时间、报文个数、报文信息列表（含流量字节数）等。

8.8 云管理的应用案例

8.8.1 M运营商私有云建设

客户简介

M运营商是一家大型网络的移动通信运营商，资产规模超过万亿元人民币，拥有全球领先的网络和客户规模。其主要经营移动话音、数据、IP电话和多媒体业务，并具有计算机互联网国际联网单位经营权和国际出入口局业务经营权。除提供基本话音业务外，其还提供传真、数据、IP电话等多种增值业务。

背景

M运营商是中国云计算的先行者和积极倡导者。2007年启动了云计算的研究项目。2011年12月正式发布了个人云服务，是国内最早进行云计算研究和实践的单位之一。M运营商在云计算研发、规划和建设以及对云计算的商业部署和推广等方面均投入巨大。云计算业务在企业内部私有云和各业务基地进行广泛的商业部署和应用，并且在多个政企客户处实

现了落地。M运营商计划建设专业的云计算中心，进一步加强云计算方面的研发与创新能力，同时不断加大投入，积极布局云计算基础设施。为满足云计算的发展，其在原有规划的广东、北京等大型基地的基础上，投入数百亿元的资金在哈尔滨、呼和浩特等地建设大规模数据中心。这些数据中心将采用模块化设计、新型供电和制冷、定制化的服务器等新技术，实现高效绿色的云计算服务。随着多地云计算数据中心的建成，M运营商加快部署、全力推动云计算的全面商用。M运营商云计算采用双云多池、集中管理、分散部署的原则进行规划。双云即公共服务云和企业私有云两大云计算平台，多池即每个云计算平台都在全国建设多个资源池，为了提高管理效率，公共服务云和企业私有云都需要建设集中的运营管理平台。

解决方案

M运营商提供统一的虚拟化和资源池管理，集成HUAWEI、IBM、HP、Fujitsu、Cisco、Juniper等8个厂商的IT设备；提供统一云管理运营平台，管理天津私有云、广东基地公有云和北京云计算资源池，实现弹性调度。

其一期建设2000台服务器规模，5PB存储规模，业务主要有内部电子邮件系统、数字档案系统、支撑网测试平台、统一知识社区、网管支撑系统、MC口信令监测系统、数据业务监测与分析系统、信息安全系统、骚扰电话监控、虚假主叫监控、手机恶意软件监控、垃圾短信监控、不良信息监控等。

M运营商选择云平台的主要特点具体如下。

- （1）高性价比：高性能，价格合理。
- （2）安全可靠：虚拟化平台支持HA、容灾、双活数据中心等能力，自动实现业务的容灾和备份，在业务的升级和例行维护中，可以有效利用虚拟化平台的热迁移功能，实现业务不中断。
- （3）运营简捷高效：可视化管理，统一集中管理，降低运营成本。

客户收益

M运营商通过实现业务平台和IT硬件资源的解耦以及在一台物理机上运

行多个虚拟机和应用，改变了一个应用独占一台服务器的低效业务烟囱模式。如此一来，其将大幅提升IT资源利用率，实现资源动态分配，缩短业务发布时间，可以帮助M运营商从容应对市场的快速变化。

- 解决原有EDC容量瓶颈，建设虚拟化弹性资源池，实现信息有效管控。
- 工程节省投资的30%。
- 业务上线由原来的一个月缩短到2天。

8.8.2 T运营商分布式数据中心

T运营商是一家面向全球提供通信、互联网等综合性服务的国际电信公司。T运营商为40多个国家的客户提供服务，是世界最大的电信运营商之一。早在2010年，T运营商已经把云化作为其战略目标，将核心业务向云计算推进，希望实现全球业务云化整合的目标。然而，经过几年的建设，云并没有给T运营商带来显著的商业机遇，传统电信业务的成本居高不下，VDI、SaaS等新型云业务基本处于停滞状态，传统的优势地位正受到竞争对手的严重挑战。

- 云业务价格过高，现今主流的云平台，如许多知名厂商都鼓吹其拥有强大的能力，能够带来高效的资源利用率。而实际上，引入云的另一面是高昂的维护和协调成本，将推高业务成本。
- 云业务产业链不完善，由于缺乏统一规划，零散的VDI、云主机或大量的SaaS业务很难形成完全产业链，商业模式的差异以及市场的变化使得云业务难以达成商业战略意义。
- 新业务上线时间过长，以一个云主机出租业务为例，从市场调研、需求分析、模型设计、IT架构部署到最后业务上线，通常要经历6~9个月的时间。
- 新业务上线需要太多时间，IT系统对问题的响应时间过慢，尤其是对跨地域、跨平台的网络问题不能及时解决。

➤ IT投入过高，据统计，每年IT投入（包括Capex和Opex）占T运营商总收入的4%以上，约20亿~23亿美元，对于电信业务为核心的运营商，IT的巨大投入导致内部成本长期处于过高的水平，阻碍了商业发展。

云没有为网络带来实际的价值，传统的带宽出租业务正在被新型的高价值业务冲击，市场空间日益萎缩，云计算被寄希望去改变这种趋势，然而当前的运营商并没有获益。T运营商从对传统IT的整合与改造出发，以云计算为出发点，大力推行云战略，但为何会呈现高开低走的局面呢？

我们从流程和技术的角度分析T运营商当前的现状。

组织流程孤立

T运营商当前的组织架构是烟囱式管理方式，从横向维度上看，每个业务系统从建设、维护到管理都是互相孤立的，如M2M、Video业务等由不同团队人员维护，商业模式和技术相对封闭；从纵向维度上看，每个国家都独立管理自身的业务，缺乏有效的沟通联动，这造成了资源的浪费，经营成本居高不下（见图8-16）。

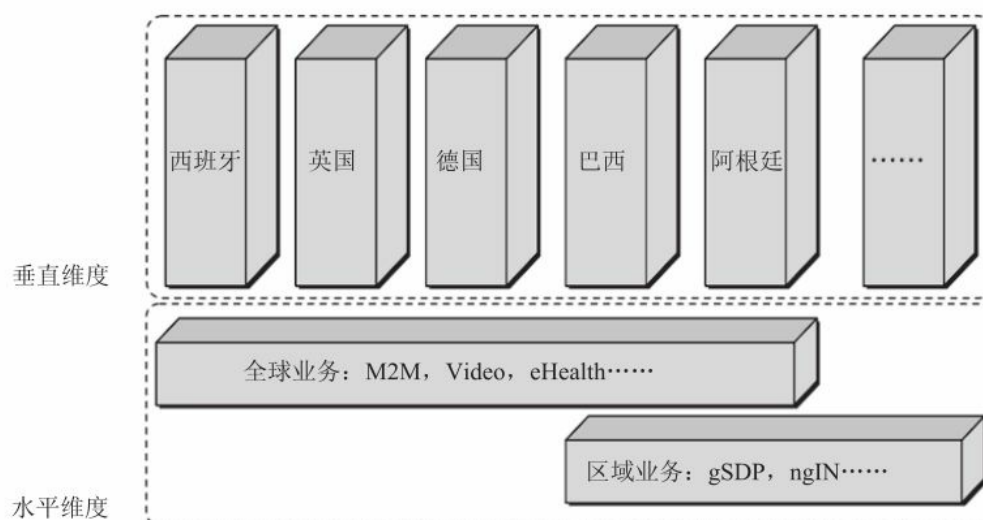


图8-16 T运营商组织架构

现代组织管理理念

只有将人力资源、各种应用和IT基础架构统一看做资源池，有序地实现跨地域、跨平台的资源调度，才能将组织的优势发挥至最大。

IT架构封闭

在电信领域，其业务系统长期处于较封闭的状态，如SGSN、MME、WAG等超过30种的业务系统依然采用定制的服务器。此外，由于不同阶段技术的演进，T运营商在欧洲、拉美等十几个国家共建设了超过27个不同规模的数据中心（机房），拥有超过15 000台不同规格的服务器，支持2 300余种应用，10多种异构数据库。图8-17为T运营商分布式云数据中心。

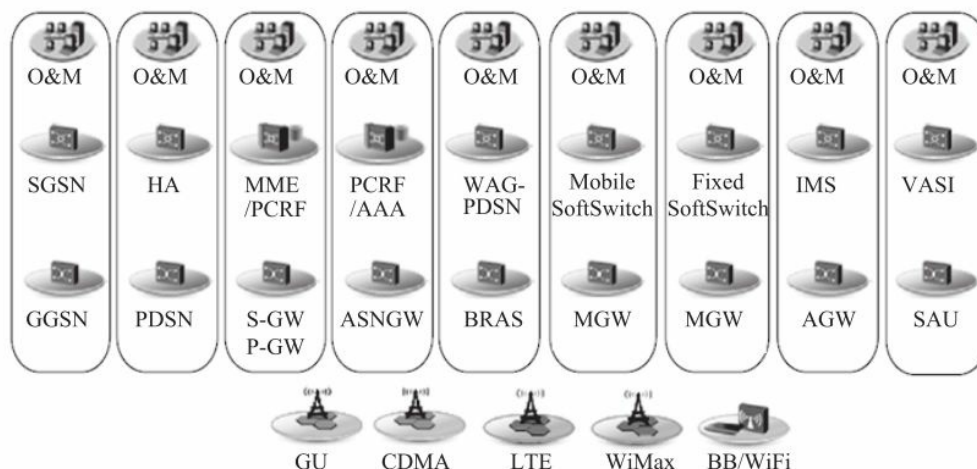


图8-17 T运营商分布式云数据中心

ALL IT ALL ONLINE战略

随着深入分析，T运营商意识到传统电信领域的困境不会因为一个个孤立的IT升级或者虚拟化等技术的引入就得到解决，相反还会带来管理复杂等新问题，因此ALL IT ALL ONLINE战略成为必然选择。

ALL IT指的是传统电信业务系统逐步x86化，首先将原有封闭定制的CT系统迁移至x86服务器上，随后采用分布式云的架构将所有资源融合成为一个“物理分散，逻辑统一”的资源池，从而实现ICT的深度融合。

ALL ONLINE指的是自动化的流程，以统一的管理平台作为流程的载体，面向内外部客户提供统一的服务入口，实现服务提供、服务保障和服务运营的三足鼎立，打破现有组织和流程的割裂格局，提升管理效

率，实现商业成功。

在此战略中，T运营采用了分布式云数据中心架构，将多个不同地域、不同阶段、不同规模的数据中心的所有资源通过逻辑集中，统一管理、统一呈现、统一运营，从而充分利用T运营商已有资源，支撑ICT服务能力高速发展。

（1）第一步：数据中心整合

根据T运营商的实际业务情况，IT基础架构需要满足其电信业务的特点。

T运营商的数据中心规划具体如下。

- 3个区域数据中心：地点位于西班牙、墨西哥和巴西，运行多种系统，连接该地区的所有业务用户。
- 16个本地数据中心：集中式架构，运行单个业务系统。

经过整合后，现仅存在两种类别数据中心，即备份与全局应用类和在线业务节点类，扁平化的结构更利于全局管理和调度。数据中心整合既保障了现网业务的高质量运行，又为后续建设资源池建设打下了基础。

（2）第二步：IT基础架构

T运营商采用分布式云数据中心的IT基础架构，将各类物理及虚拟化基础设施、软硬件资源、人力资源统一看做资源池进行管理及应用，实现跨数据中心的资源调度。

- 分布式和虚拟化是核心

其针对T运营商跨地域的业务模型、分布式的IT架构，拉通多个站点的计算、存储与网络资源弹性调度，提供对资源请求者相对透明的最优化资源利用和调度。

- 智能网络方案是基础

其将网络设备控制面与数据面分离开来，以网络操作系统的概念使底层

网络设备的具体细节抽象化，同时还为上层应用提供统一的管理视图和编程接口。这种智能网络解决方案让电信业务只需要对网络资源提出需求，无需关心底层网络的物理拓扑结构，为核心网络及应用的创新提供良好的平台（见图8-18）。

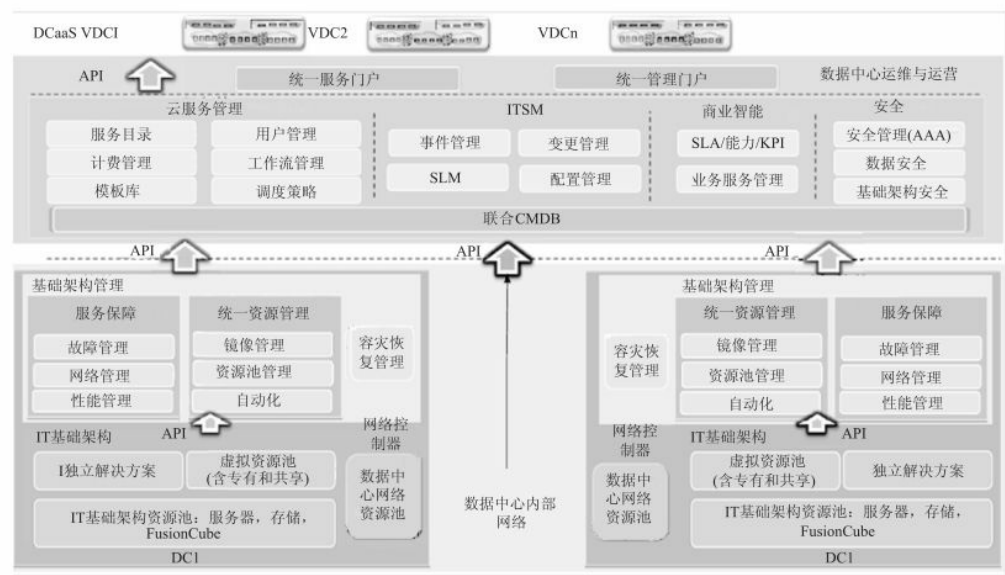


图8-18 智能网络方案

➤ 统一管理解决方案是灵魂

对于跨多个云数据中心站点的所有资源，其实现全生命周期内（包括自动化发现与配置、管理软件安装部署、资源发放与回收、业务应用软件部署、资源实时占用状态监控、告警故障等）的全方位的管理功能。

（3）第三步：一切皆为服务

虚拟化的数据中心已经就绪，跨数据中心的网络管道也具备智能调度的功能，统一对外提供服务成为T运营商接下来的核心任务。

统一的服务门户和开放的API将资源池转变成一个服务中心，定义和公开服务目录，让内外客户按需申请、使用和退订，并精确地对资源进行计量。此外，平台还提供自助服务门户，对用户设置不同的角色，根据角色对用户进行权限控制。用户通过一个统一的门户网站自行选择服务目录，快速部署业务和运维系统，增强管理效率。

➤ 区域服务经理：负责管理所有的数据中心（DC，Data Center），创建和维护服务水平协议。

➤ 本地服务经理：支持本地数据中心的运维操作，并提供本地资源信息。

➤ 虚拟数据中心服务经理：对每个VDC（Virtual Data Center，虚拟数据中心）进行服务支持，保障服务质量，监控和维护管理工具。

（4）第四步：服务自动化

服务的自动化是T运营商商业成功的保障。其采用Orchestration软件对运营商执行的日常任务和职能进行编排。其将协调服务器、客户端和网络设备的自动化解决方案，从而更快地配置资源。服务的自动化将实现跨多个应用程序和工具，以及跨多个IT群体的操作自动化，从而使跨数据中心的业务互联互通变得更容易、更快、更可靠。这里流程的基本定义是至关重要的，因为T运营商的物理数据中心和虚拟数据中心内可能都是租户，需要在多租户环境下提供IT资源，按需分配，利用流程来保障分配过程的有序和可管理。

此外，服务自动化还包含计费管理、资产和配置管理、软件许可管理、变更和发布管理、服务台、服务水平管理系统等模块，实现自动化服务管理，形成深入、智能的分布式云数据中心。

T运营商收益

T运营商借助本项目，规划出未来电信数据中心的架构设想，其最大的收益在于制度标准和规范的制定，其可供全球24个子网使用（见图8-19）。

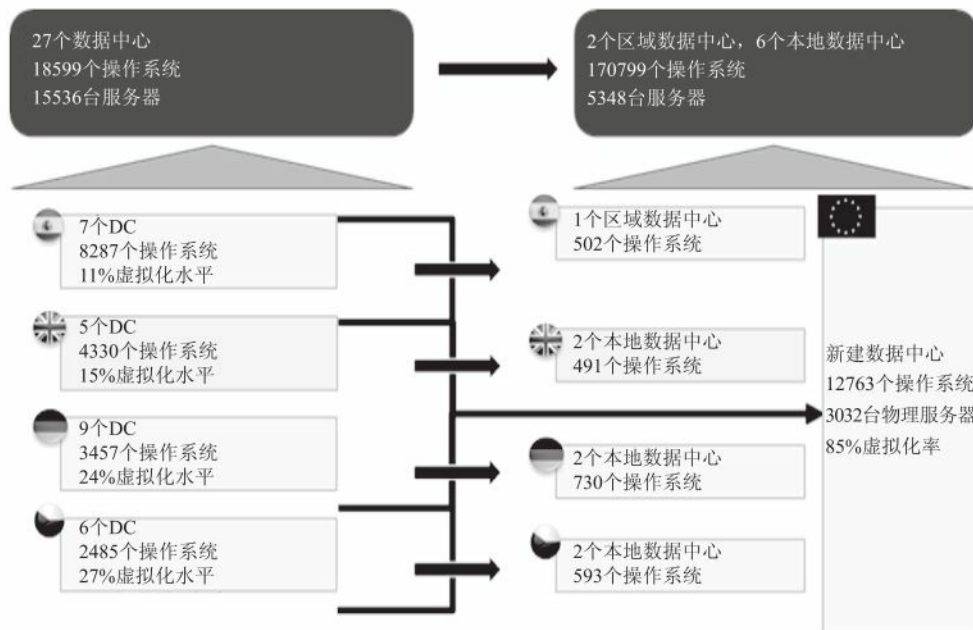


图8-19 T运营商的收益

其具有以下两个优点。

- **降低成本：**分布式云数据中心基础架构以标准化组件、简化基础架构使得所有的软、硬件设施和人力资源被看做是一个统一的“逻辑资源池”，通过全局拉通来进行管理调度。其从根本上解决了烟囱式的投资模式，实现高投资回报率。
- **提升效率：**以业务为中心实现流程标准化、部署模板化、服务自动化。目前总部可以实现对全球30多个不同地区的数据中心进行实时管理，部署新业务，全球数据中心统一入口，统一呈现，统一管理，这样大大地简化了管理，IT费用占收入比例显著下降。

8.8.3 新加坡S运营商中小企业IT应用托管

新加坡政府历来对信息产业极为关注，在其推动下，新加坡已经成为亚太地区的电子商务中心。为强化该地位，新加坡政府启动了国家宽带建设，并随之推出“政务云”服务计划，旨在进一步推动信息化发展。而新加坡S电信运营商希望借此契机，在云计算带来的变革中进一步优化产业结构，推出公有云服务，谱写云篇章。

紧握时机

S运营商具备众多优势，同时可带来更大的外部影响。

首先，S运营商有建立公有云服务的内部优势：拥有自有的数据中心和网络，这正是云计算的基石；拥有供电、制冷、机房等空余资源可以进行利旧应用，创造新价值；借助新加坡国家宽带战略推动，S运营商网络覆盖已达20 000座楼宇，可实现对绝大部分中小企业的覆盖，而服务中小企业正符合新加坡的国家战略需要。

其次，S运营商率先建立公有云服务将带来更多的外部影响。IT企业和CT企业在云计算领域并无太大差距，快速部署公有云服务可获得进入云计算市场的先发优势；在新兴市场领域，率先实现业务应用部署，有利于品牌建设；此外，适逢新加坡开展“政务云”计划，S运营商通过公有云建设，可以发挥该计划的试点作用，增加服务国家的机会。

因此，从自身优势到外部环境来看，S运营商建立公有云服务的时机已经到来。所以，S运营商高层极为重视，均表示将利用国家宽带规划和政府推动信息产业发展的机会，进入中小企业市场，同时扩大业务。

选择合作伙伴

从全球来看，运营商建立公有云服务的成功案例为数不多，而市场需求容不得长时间的思考。因此，S运营商需要快速探索公有云服务的建设和运营模式。基于此，S运营商对该项目的合作伙伴提出了以下要求：必须具备快速集成交付能力，可以实现设备快速到货并完成安装测试；必须具备快速定制开发能力，解决方案可以根据S运营商的需求量身定做；必须拥有成熟的解决方案和长期投入的决心；在ICT融合转型过程中积累丰厚的经验，可以更好地帮助运营商转型。

创新设计

S运营商公有云服务运营平台建设的第一阶段主要是提供基础设施服务。借助S运营商已有的机房、供电、制冷等设施，结合华为提供的服务器、网络、存储及安全设备，同时依靠华为自研的FusionSphere云操作系统，S运营商和华为快速完成了公有云服务的架构搭建，以提供大规模、高性能的虚拟资源服务。

为保证该项目快速成功上线，S运营商和华为在运营管理方面做了很多创新的方案设计，尤其是系统安全保障方面的创新。如云主机服务商限制用户最多创建20个虚拟机，来防止恶意用户瞬时创建大量虚拟机，造成系统瞬时负载过重。但这种限制只对大规格使用者有意义，对使用小规格虚拟机、但数量要求多的使用者，该限制很不合理。当用户申请特殊规格虚拟机时，这种方案必须变更。因此，该方案虽可实现对系统的保护，但灵活性不足。经过深入探讨，S运营商和华为确定通过对用户Core（核）、Memory（内存）、Disk（硬盘）三个维度进行限制，任一维度超出限制即不能创建虚拟机，从根源上保证系统的安全运行，同时可以保证使用小规格虚拟机用户可以申请更多的虚拟机，确保将来在规格限制内申请特殊规格虚拟机而不用变更方案。

在商业模式上，为实现快速盈利，S运营商和合作伙伴结合各自的成功经验，共同探讨出一套灵活的销售模式。不同于其他企业不区分用户类型，统一按照1个月进行试用的模式，S运营商对试用账户进行了多种区分，既可以让普通用户充分体验S运营商公有云带来的良好服务，又可以给重点用户提供试用期更长、资源更多的服务，以提高用户粘性。在资费方面，S运营商也采用了灵活的方式，如针对散户、不同套餐用户、VIP用户等多种不同类型用户制定了不同的价格标准，同时用户在将试用账户转正式账户时，可随意变更套餐、终止套餐或启用新套餐，实现了虚拟资源价值的最大化，方便用户使用。

扩大服务

目前，S运营商的IaaS业务已经率先正式商用。凭借成熟可靠的解决方案和运营模式，S运营商已经吸引了包括埃森哲及新加坡政府等高端客户业务的大规模迁移和入驻，这给S运营商的云计算服务带来了极佳的品牌效应。如今，不少中小企业客户已经或正在考虑租赁S运营商的虚拟资源，为S运营商的云计算服务实现快速、规模化运营和赢利打下良好基础。

但是，从未来商业运营模式分析，仅有硬件基础设施的业务和服务不足以在云计算领域获得更大的发展，还必须提供更多、更优质的软件应用服务来协助中小企业在运营商创建的沃土上快速成长，形成繁荣的生态链，而强大的聚合平台正是构建生态链中最关键的一环。

面对中小企业市场低成本、灵活、多样等多重特点，功能强大的聚合平台必不可少，其催生出的软件运营模式在欧美等IT业发达地区已经取得

了良好的发展。而在新加坡，软件应用服务却是新生事物。新加坡有超过18万家企业，其中99%为中小型企业。对于这些企业来说，一方面，他们面临资金短缺、技术支持人员流失严重的现状；另一方面，又急需专业的IT系统和服务帮助其降低运营成本，增强核心竞争能力。

软件应用服务正是解决这些矛盾的最佳途径。用户可以根据自己的应用需要从服务提供商那里订购相应的软件应用服务，并且可以根据企业发展的变化，调整所使用的服务内容，具有很强的伸缩性和扩展性，同时这些应用服务所需要的专业维护与技术支持也都是由服务商的专业人员来承担的。

S运营商通过合作伙伴的协助，在公有云基础设施（IaaS）服务平台为新加坡政府进一步推动信息化发展和助力中小企业发展做出贡献。在云计算的广阔领域中，任何企业都不可能独自实现全领域的开发建设。只有合作，才能共赢，只有共谋，才能发展。

第9章 云安全架构与应用实践

云计算的“安全”实际上与我们通常意义上的“安全”概念上是基本一样的。

以我们最关注的个人安全为例进行类比可以看到，个人安全威胁可能会涉及个人财产损失、个人人身伤害、个人隐私侵犯等。针对这些可能的个人安全威胁，人类很早就开始穿上衣服来遮挡隐私；然后盖上房子修建围栏来防止野兽的侵扰；然后房子又加上锁来防止同类来偷窃；到了现代又发明了保险柜，防止小偷撬锁；后来又把钱存到银行防止小偷连保险柜一起偷走……而进入云计算时代，个人安全威胁以新的形式以及更大的广度在扩大，例如将照片、视频、通讯录、私人日记放在了网盘的个人空间里（云），一些担心就会随之而来，如网盘信息会不会被人贩卖？托管在公有云上的客户关系管理系统的客户数据信息会不会被竞争对手拿到？业务系统信息是否可能被篡改？可以看到，在云计算时代，为了保障个人安全，我们只盖个房子、加个锁是远远不够的，而且房子怎么盖，锁加在哪里都成了问题。这时，我们就需要使用系统工程的方法来构筑个人安全的防范问题。

对应地，在云计算领域，使用系统工程的方法来建立和完善云计算体系的安全，这便是“端到端云安全架构”。

9.1 端到端云安全架构

9.1.1 云计算中的主要安全威胁

概括地讲，云计算的主要安全威胁仅4个字，即“天灾人祸”：

- “天灾”泛指各种不可抗力，例如地震、火灾、水灾等；
- “人祸”指得某些个人或者组织为了实现其利益而对其他人或其他组织尽其所能地进行入侵、攻击、窃取、破坏等行为。“人祸”是云计算的主要安全威胁。

云计算体系可能遭受的威胁来自多个层次。

网络层次

- 数据传输过程中的数据私密性与完整性存在威胁：目前多数用户仍使用HTTP方式（未加密）而非HTTPS（加密）访问云资源。一些敏感信息如密码，可能被窃取。
- 更容易遭受网络攻击：云计算必须基于随时可以接入的网络，便于用户通过网络接入，方便地使用云计算资源。云计算资源的分布式部署使路由、域名配置更加复杂，更容易遭受网络攻击，比如DNS攻击和DDoS攻击。对于IaaS，DDoS攻击不仅来自外部网络，也容易来自内部网络。
- 资源共享风险：云计算的共享计算资源带来的更大的风险，包括隔离措施不当造成的用户数据泄漏、用户遭受相同物理环境下的其他恶意用户攻击；网络防火墙、IPS虚拟化能力不足，导致已建立的静态网络分区与隔离模型不能满足动态资源共享需求。

虚拟化层次

- Hypervisor的安全威胁：Hypervisor为虚拟化的核心，可以捕获CPU指令，为指令访问硬件控制器和外设充当中介，协调所有的资源分配，运行在比操作系统特权还高的最高优先级上。一旦Hypervisor被攻击破解，在Hypervisor上的所有虚拟机将无任何安全保障，直接处于攻击之下。
- 虚拟机的安全威胁：虚拟机动态地被创建、被迁移，虚拟机的安全措施必须相应地自动创建、自动迁移。虚拟机没有安全措施或安全措施没有自动创建时，容易导致接入和管理虚拟机的密钥被盗、未及时打补丁的服务（FTP、SSH等）遭受攻击、弱密码或者无密码的账号被盗用、没有主机防火墙保护的系统遭受攻击。

数据与存储威胁

- 静态数据的安全威胁：静态数据可以加密保存，如简单对象存储业务，用户通过客户端加密数据，然后将数据存储到公有云中，用户的数据加密密钥保存在客户端，云端无法获取密钥并对数据进行解密。

➤ 数据处理过程的安全威胁：数据在云中处理，数据是不加密的，可能被其他用户、管理员或者操作员获取。

➤ 剩余数据保护：用户退租虚拟机后，该用户的数据就变成剩余数据，存放剩余数据的空间可以被释放给其他用户，如果数据没有经过处理，其他用户可能获取到原来用户的私密信息。

身份认证与接入管理

云计算支持海量的用户认证与接入，对用户的身份认证和接入管理必须完全自动化，为提高认证接入管理的体验，需要云简化用户的认证过程，比如提供云内所有业务统一的单点登录与权限管理。

图9-1 描述了云计算管理员、用户和黑客对云管端所造成的威胁。

以上列出的仅仅是来自某个个人的可能威胁，还未列出来自某个组织（叫“团伙”可能更容易理解）的威胁，比如某个大型公司内部的某个小团伙对某大型公司进行破坏活动，那么这个破坏力会成倍增长，因为这个团伙成员可能来自管理层、IT、内外部用户和黑客。

模块	威胁源	管理员	用户	黑客
端	TC	非法操作：如利用TCM与TC间正常的升级通道，植入木马控制TC 伪造非法TCM：控制TC 权限滥用：对TCUSB端口管理不合理	恶意用户：仿冒其他用户登录，破解密码 非法TC：非法TC具有获取VM数据的能力	伪造TCM管理员 TCM漏洞攻击 TC被非法破坏：植入木马，非法获取VM数据
	SC	权限滥用	数据泄露到本地，如截屏	SC漏洞攻击
管	网络		截获其他用户密码 PC等设备绕过安全网关	常见网络攻击
云	虚拟机	非法重置用户密码 误挂卷 利用虚拟机备份文件非法恢复用户数据 虚拟机自然损坏	用户非法登录：弱口令或口令保管不善 用户虚拟机被篡改 攻击相邻虚拟机，如ARP攻击 非授权访问相邻虚拟机 攻击虚拟化平台 利用虚拟化资源从事非法活动，如攻击外网 虚拟机迁移过程中安全策略失效	类似PC的常见攻击
	虚拟化层	管理员非法登录：利用弱口令或口令保管不善 权限滥用：在缺少三权分立场景下易发生 关键操作无法回溯 破坏镜像文件，植入木马 管理员权限扩大化：如节点间采用互信，则获取单节点权限即可控制整朵云 非法获取敏感信息，如数据库口令 非法监视用户虚拟机流量 非法获取用户密码	还原出前一用户硬盘数据 还原出前一用户内存数据	利用租用的虚拟机攻击虚拟化平台 利用租用的虚拟机攻击虚拟化管理平台，如利用OS/Web漏洞 虚拟机迁移中截获用户数据

图9-1 不同客户的云安全威胁

9.1.2 端对端的安全架构

上面分析了云计算环境中所面临的各种威胁，不同级别的威胁，其相应

的安全保障措施是不同的。企业或组织对安全级别要求越高，这个端到端安全架构的价值也就越大，安全架构之下的具体技术保障手段也更严谨和丰富，如图9-2所示。

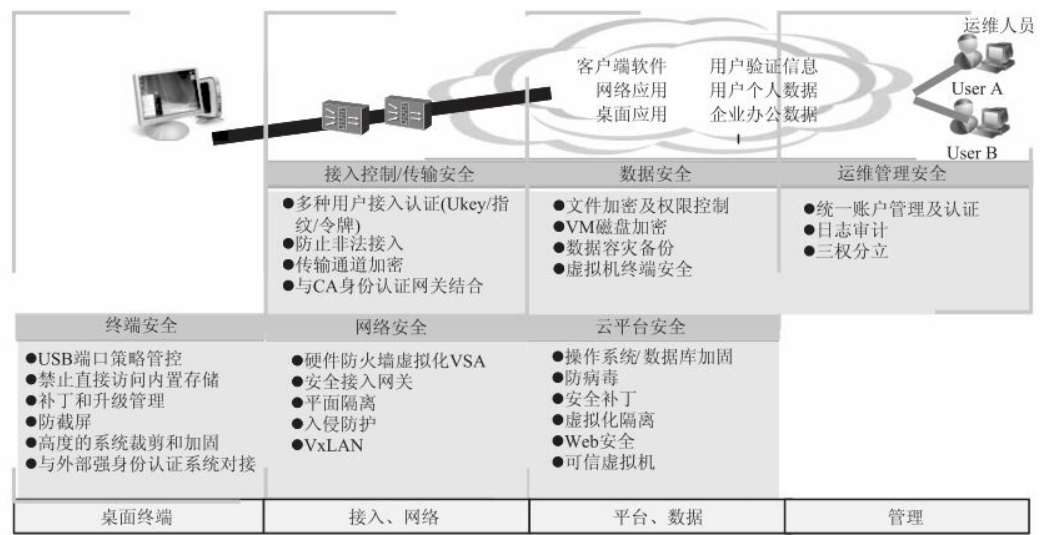


图9-2 云计算E2E安全架构

结合云计算的风险，端对端云计算安全架构有以下几个核心部分。

“云终端”安全

采取USB端口策略管控、禁止直接访问内置存储、补丁和升级关联等。

其实质是采用专用的TC（Thin Client，瘦客户端）终端，这种TC终端用于显示云中心的图像，是一个专用的嵌入式系统。TC这种瘦客户端很像家里安装的机顶盒，看到的节目（TC对应的内容）都来自电视台（TC对应的云平台）。和机顶盒不同的是，机顶盒使用遥控器进行操作，TC可以直接连接键盘鼠标，像以前使用PC那样（跟PC使用体验基本一致，只是受到了更严格的安全管制）。采用这种TC终端能够实现较好的安全性，包括以下几个方面。

➤ BIOS仅从内置存储引导

TC的BIOS只保留从内置存储引导的方式，没有保留其他的引导方式，如USB引导、PXE（Preboot Execute Environment，预启动执行环境，网络远程启动）引导。

➤ OS安全

TC是一个精简的、封闭的Linux/WES7/XPe操作系统。

- **禁止存储类设备使用：**TC的操作系统从驱动层可禁止USB存储设备的使用，不包含任何USB存储的驱动。包括U盘、USB光驱等在内的USB存储设备无法在TC本地使用。
- **禁止直接访问内置存储：**操作系统的TC在本地没有暴露内置存储访问接口，用户只能通过系统提供的程序间接访问，这样能有效地避免系统文件被破坏。
- **禁止任意安装程序：**操作系统的TC在本地不提供安装软件的接口，用户或者其他第三方人员无法自行安装任何软件。如果需要在TC上安装软件，则只能通过TCM（专门的TC管理系统）将软件包下发到TC上，然后在TC上进行安装。

简单理解“云终端安全”的目标，那就是，对安全可能构成威胁的操作基本都被禁止。

“管道”安全

管道安全包括接入安全和网络安全。

接入安全包括多种方式的接入认证（如USB KEY/指纹/令牌等）、防非法接入（基于网络IP参数和用户ID）、传输通道加密、与CA系统对接等。

网络安全包括防火墙的访问控制（这里的防火墙除传统防火墙以外，还包括虚拟化的防火墙等多种类型）、安全接入网关（用于与TC进行认证，建立安全通道，甚至提供负载均衡的功能）、网络平面隔离（提供业务网、管理网和存储网的三网隔离）、网络入侵防护（如提供防拒绝服务攻击功能）、基于VxLAN的网络VLAN划分等。

理解“管道安全”的重要性，有一个形象的例子。换位思考一下，一伙劫匪，他们为了钱，是抢银行容易呢，还是抢运钞车容易？答案肯定是运钞车，因为运钞车在路上，再怎么防护，也没有银行防护得严密。在银行内部，通往金库是最薄弱的环节，也是通路。那么为了安全防护，在

外部，要考虑运钞车本身的坚固性、押解人员以及行进路线的安全性；在银行内部，通往金库的道路上要层层设卡，设立各种监控和密码设施。

“云”安全

云安全包括云端数据安全，这是云计算重点需要考虑的安全内容。云计算中的数据太过集中会造成用户的担忧，信息资产的集中存储也是网络攻击的重点所在。“云”安全包括云数据安全和云平台安全两部分。

云数据安全的解决方案包括文件加密及权限控制、VM的磁盘加密、数据的容灾备份、虚拟机终端安全。

云平台安全包括云操作系统和数据库的安全加固、防病毒、安全补丁、虚拟化隔离、Web安全、可信虚拟机。

其中的虚拟化隔离和可信虚拟机（借助可信芯片）是下面重点介绍的内容。

“管理”安全

管理安全包括统一账号管理及认证，日志审计，三权分立等内容。

下面对云计算中的重要安全技术及实现方案进行介绍。

9.2 可信计算TPM/vTPM

TPM英文全称为Trusted Platform Module，即可信赖平台模块；vTPM的英文全称为Virtualizing the Trusted Platform Module，即虚拟化可信赖平台模块。

在云计算环境中，用户失去了对虚拟机的完全控制，导致用户无法信任虚拟机环境，这成为了用户部署云计算的一个重要障碍。因此采用可信计算和虚拟化技术的结合成为热点。基于TPM/TCM/TXT可信硬件技术，在云平台上开发可信虚拟机原型系统，可实现虚拟机的可信启动、可信运行、虚拟机可信迁移等，从而为虚拟机构建一个可信的计算环境，提升用户对虚拟机环境的信任度。

虚拟机的可信是高等级信息系统中的关键点，主要包括：

- 防止节点的软件配置、虚拟机的OS被黑客攻击篡改；
- 虚拟机防止被其他虚拟机攻击、嗅探；
- 防止虚拟机被恶意的人员启动和利用；
- 防止虚拟机迁移到不被认可的非安全Host主机上；
- 防止Dom0的恶意管理员利用特权账户发起对DomU的监控与攻击（Dom全称Domain，是CPU虚拟化内核调度中的一个名词，简单理解，CPU中有很多Domain，每个Domain中存放一个操作系统软件，Domain0中放的是虚拟化软件内核操作系统软件，Domain0中的操作系统软件有比其他Domain更高的CPU指令执行特权，恶意管理员可能利用这个特权去监控其他Domain中的操作系统）。

虚拟机可信计算技术是当前虚拟化环境中的技术发展热点，也是技术难点，一般通过可信芯片技术来实现。技术实现方案如图9-3所示。

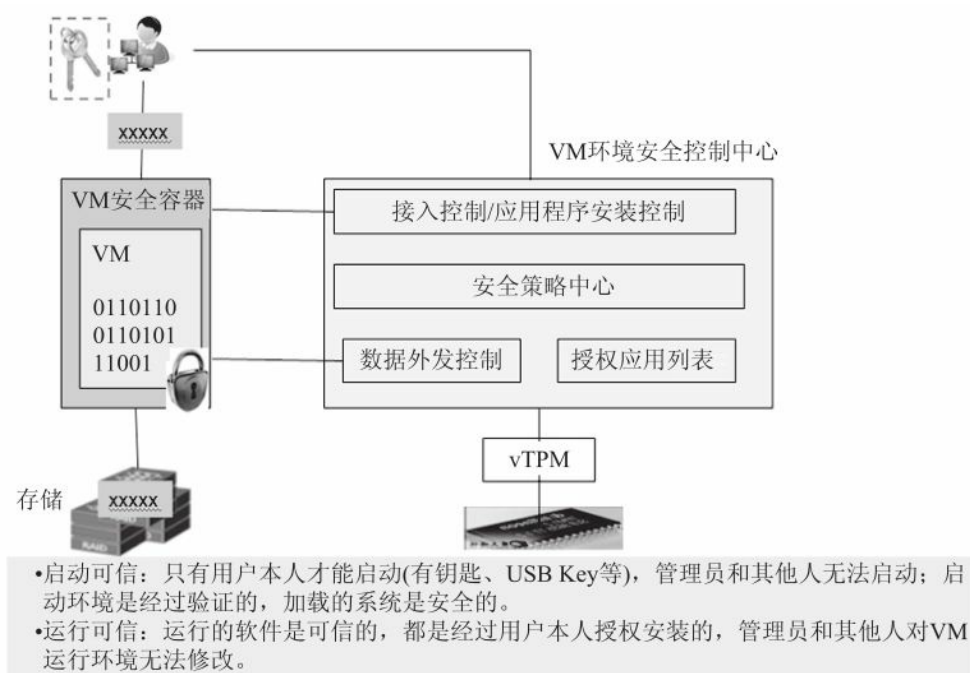


图9-3 可信芯片技术原理

图9-4 是实现虚拟机的可信启动原理。

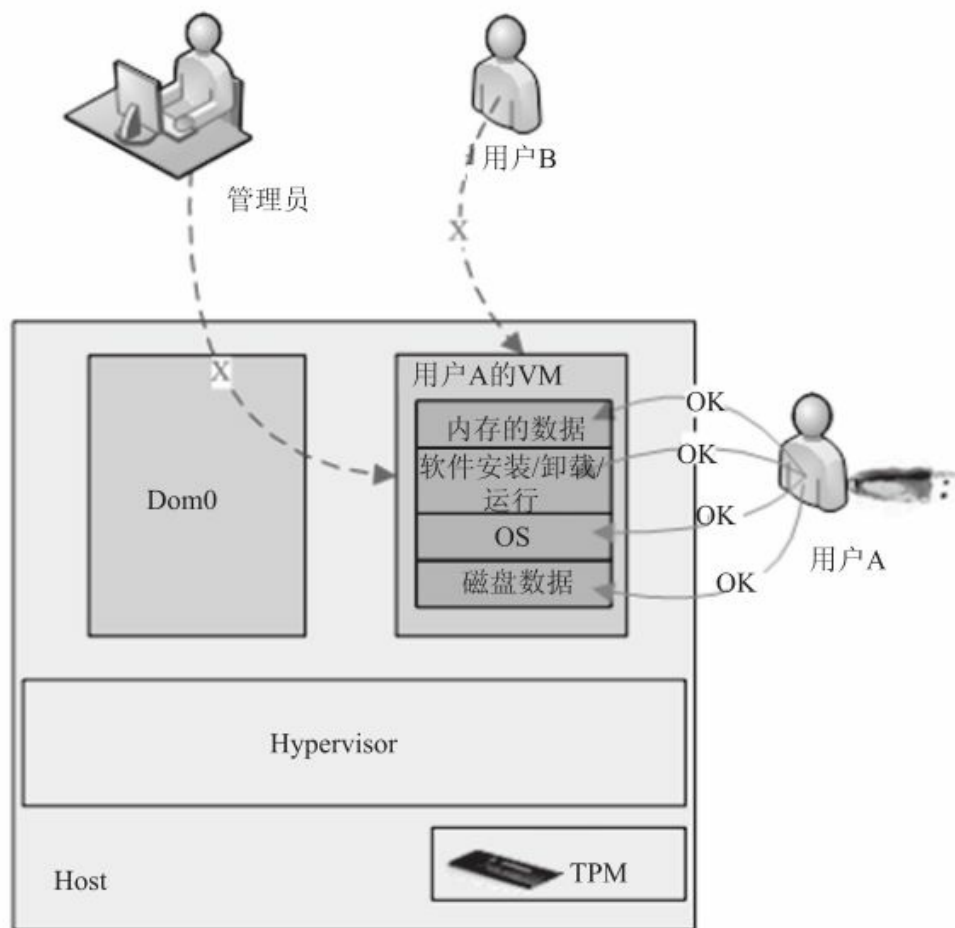


图9-4 可信启动原理

用户的VM初始内存的数据（不变的部分）、软件安装/卸载/运行信息、OS信息、磁盘数据以Hash的方式做了运算，成为度量的基数。该基数保持在TPM芯片中，后续的度量值与之比较。（简单理解，Hash是一种算法，能够对每个目标状态以唯一数值的方式进行标注，相当于你家养的牲畜都被它盖上戳，多了少了都可以看出来，有狼披着羊皮混在你的牲畜圈也能看出来，或根本就混不进去）。

其实现的功能是：

- 可信VM的OS状态受到TPM的保护，防止OS被修改；
- 可信VM中软件的安装/卸载/运行需要得到用户授权，并可防止

软件被修改；

➤ 监控Dom0和管理员对可信VM的内存访问，防止VM内存数据被获取。

用于云计算当中的TPM芯片可能有几种安装形式：内嵌到主板之上、或以插卡的形式安装到主板上的预留插槽。比如华为的Tecal系列服务器，全部采用TPM插卡形式，TPM插卡作为服务器的选配件提供。

下面几个小节介绍一下TPM可以实现的主要功能。

9.2.1 TPM功能1：主机启动/静态度量

基于上述可信技术可以实现许多非常有用的功能，举例来讲，云计算环境中主机的安全是云计算安全的基础，如果主机操作系统被篡改或植入恶意代码，那么在主机之上运行的云操作系统都变得不安全。基于TPM技术，可以把合法的主机信息保存起来，在系统启动过程中进行有效判断（见图9-5）。

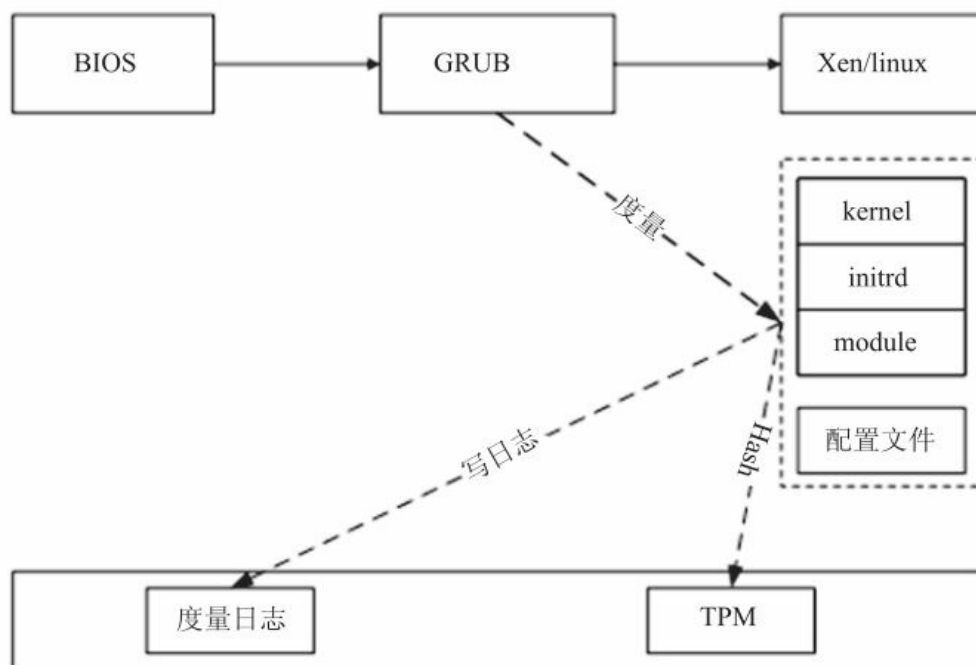


图9-5 主机静态度量流程

我们将实现这一功能的过程描述如下：

- BIOS将控制权交给GRUB；
- 在kernel、initrd、module命令中加载操作系统内核镜像及其他模块，GRUB引导操作系统启动，同时进行度量，将hash值扩展到TPM PCR10（PCR、Platform Configuration Register、平台配置寄存器），将记录写入度量日志；
- 度量配置文件列表中的文件，将hash值扩展到TPM PCR10，将记录写入度量日志；
- 执行boot命令，操作系统启动；
- 在操作系统启动后，传值模块自启动，将主机静态度量信息传出。

9.2.2 TPM功能2：虚拟机的静态度量

同样，基于TPM，可以对虚拟机的合法性进行保护。主要原理是在虚拟机启动前，依据云管理服务器下发的静态度量配置文件构造度量文件列表，然后将虚拟机镜像mount放在指定目录，mount的镜像分为Windows和Linux两种操作系统。

根据度量文件列表依次对查找到的文件进行SHA-1度量操作，得出160位hash值，同时将每一个文件的hash值进行迭代操作，得到最终的度量结果，将该度量结果与各个文件的hash值构成的日志文件上传给上层，为虚拟机镜像是否遭到篡改提供依据。

9.2.3 TPM功能3：主机动态度量

TPM除了用于对操作系统及其虚拟机系统的静态信息的保护以外，在系统运行中，还能根据系统的运行状态信息进行动态保护。主机动态度量分为两个模块：Xen度量模块和Dom0度量模块（见图9-6）。



图9-6 主机动态度量原理

Xen度量模块部署在UEFI中，利用SMM机制进行实时度量保护，主要在UEFI（新型UEFI，全称“统一的可扩展固件接口”，英文为Unified Extensible Firmware Interface，是一种详细描述类型接口的标准）中进行实现，借助CMOS来传递度量的地址和范围，可以进行手动和定时两种触发方式，当度量程序被触发后，会对CMOS中指定的数据进行度量，并将度量值扩展到TPM（Trusted Platform Module）的PCR（Platform Configuration Register）中。手动触发是通过管理员在Domain0中输入触发指令来触发SMI Measurement Handler，而定时触发是由UEFI中的定时协议提供的功能，到达指定时间会自动执行SMI Measurement Handler；SMI Measurement Handler会从CMOS中读取指定的度量范围和地址并完成度量操作，最后将度量结果生成日志信息。

Dom0度量模块部署在Xen中，主要对上层Domain0中的GDT（Global Descriptor Table）、IDT（Interrupt Descriptor Table）和模块及驱动进行度量，并负责主动获取在内存中Xen的度量值，然后生成度量日志文件。

9.2.4 TPM功能4: VM动态度量

如图9-7所示，本部分为动态度量结构图，分为调度模块和度量模块。度量模块负责对虚拟域进行度量，调度模块负责Domain-0与度量模块的通信。

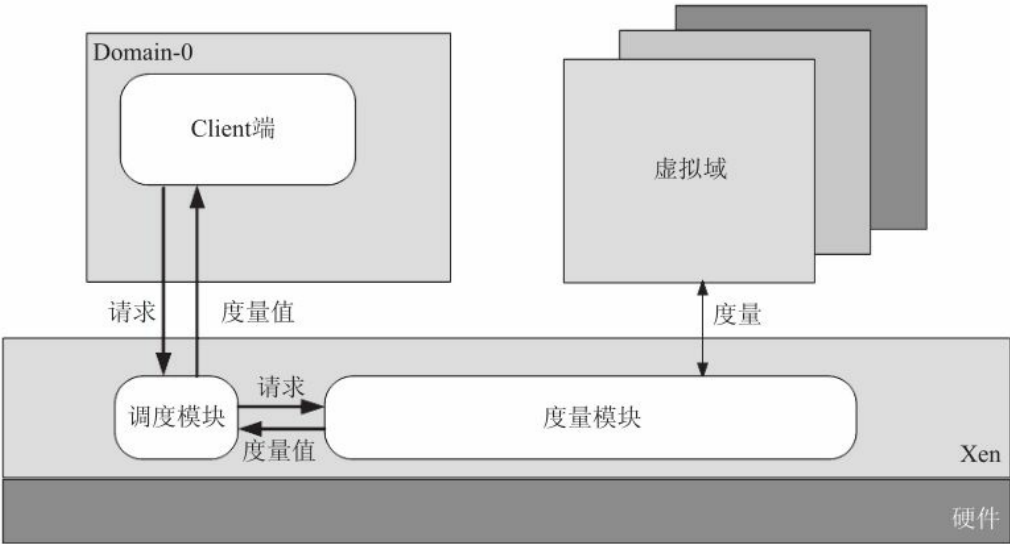


图9-7 VM动态度量流程

通信包括两个方面：第一，将度量模块度量得到的度量值传送到Domain-0的Client端；第二，接收Client端发来的命令请求，并将请求发给度量模块，对度量模块的行为进行触发。度量分为定时度量和手动度量两种。手动度量由Client触发，度量模块接收到手动度量请求后，完成度量行为，并将度量值回发给Client端。定时度量的触发由度量模块中的timer完成。每一个虚拟域对应一个timer，Client端能够设置每个timer的时间片以及是否进行度量。对于不同的虚拟域，其有不同的度量点，如表9-1所示。

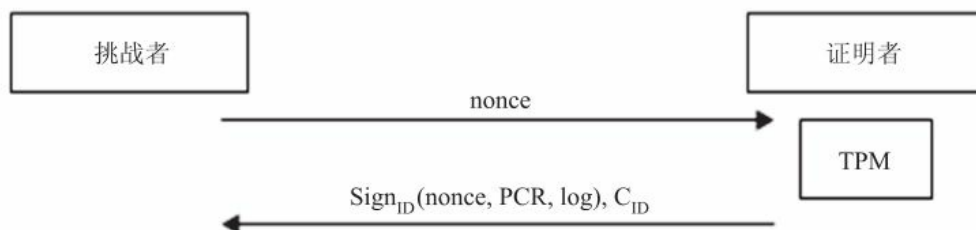
表9-1 虚拟域的不同度量点

虚拟机类型	度量点
Windows	GDT IDT SSDT驱动
Linux-hvm	GDT IDT KERNEL内核模块

9.2.5 TPM功能5: 远程证明

远程证明是对平台做全面的度量，向远程通信方证明自身运行环境是可信的。远程证明是一个综合完整性校验和身份鉴别的过程，同时向验证者提供一份可信的平台状态报告。TPM是报告的可信根，能够保证对当前完整性度量值做可信的报告。

远程证明是通过远程证明协议来实现的，如图9-8所示。



注：PCR英文全称为Platform Configuration Register，即平台配置寄存器PCR。

图9-8 远程证明协议

我们基于远程证明协议（见图9-9），根据不同的场景，收集客户端相应的度量值和度量日志，开始进行完整性检查。服务器产生一个随机数，发送给客户端。配备TPM的客户端对度量值用SHA1算法求出哈希值，再用生成的签名私钥对其进行签名，形成完整性报告，报告包括主机名、随机数、度量值、哈希值、签名及一些平台相关信息等。最后，将完整性报告和度量日志一起发送给服务器。服务器检查随机数、验证哈希、验证签名，通过比较签名值和基线值判断平台是否可信。

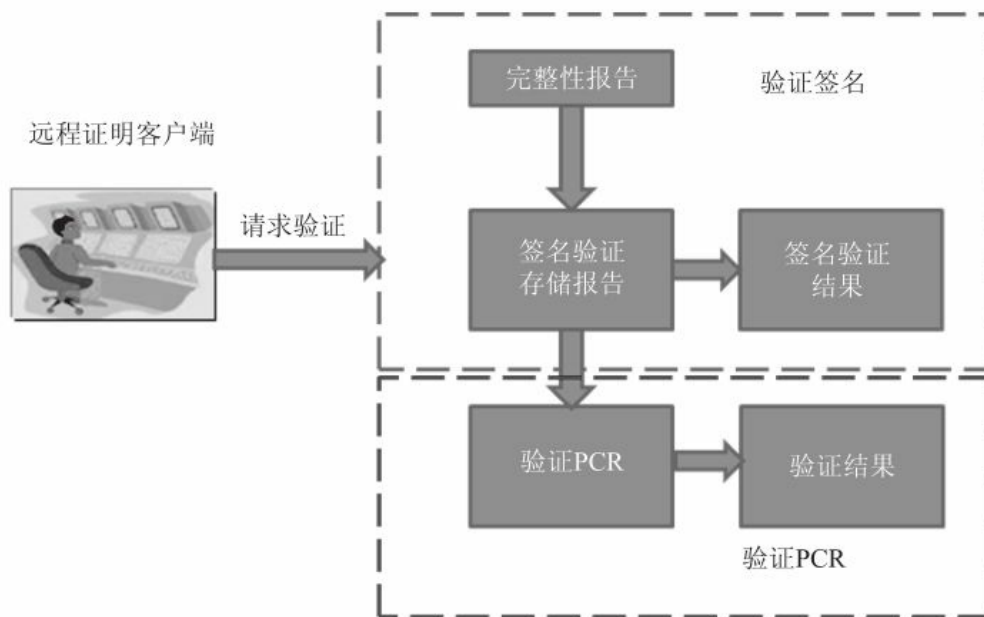


图9-9 远程证明流程

9.3 虚拟机的安全隔离

9.3.1 vCPU调度隔离安全

Hypervisor普遍采用了硬件辅助虚拟化技术（以下以服务器最常用的Intel CPU硬件虚拟化技术VT-x介绍），VT-x将CPU的操作扩展为两个forms（窗体）：VMX Root Operation（根虚拟化操作）和VMX Non-root Operation（非根虚拟化操作），VMX Root Operation设计供Hypervisor使用，其行为与传统的IA32并无特别不同，而VMX Non-root Operation则是另一个处在VMM（Virtual Machine Monitor）控制之下的IA32环境。所有的forms都能支持所有的Ring0~Ring3共4个特权级，这样在VMX Non-root Operation环境下运行的虚拟机就能完全地利用Ring 0等级（见图9-10）。

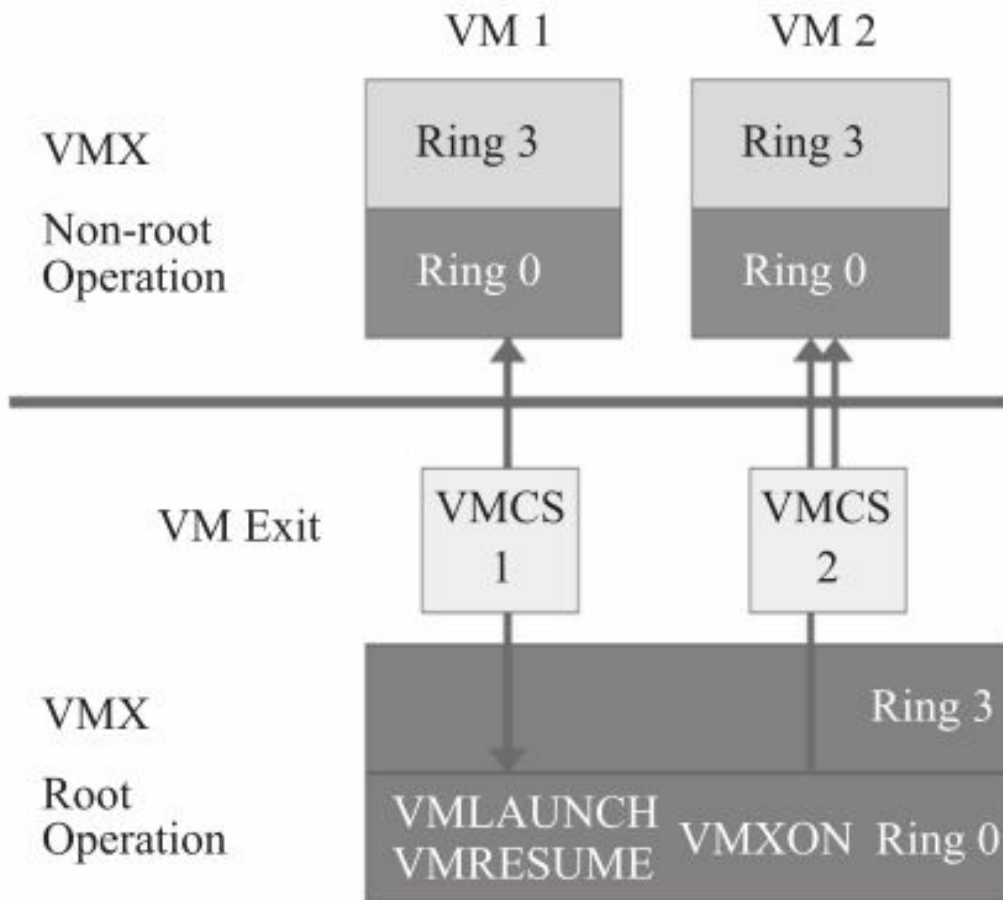


图9-10 CPU的隔离原理

Root Operation环境和Non-root Operation环境之间可以切换。如果虚拟机要进行一些特权操作，比如I/O访问、对控制寄存器的操作、MSR的读写指令等，此时就进入了Root Operation环境，当处理完这些特权操作后，将重新进入Non-root Operation环境继续虚拟机的执行。从Root Operation环境到Non-root Operation环境叫VM Entry，反之称为VM Exit。同时，VMM可以通过执行VMXON和VMXOFF指令打开和关闭VT-x。

在Root Operation环境和Non-root Operation环境之间进行切换时，有一个虚拟机控制结构VMCS（Virtual Machine Control Structure）进行控制和管理，当虚拟机被创建时，VMM就同时为每一个vCPU创建一个VMCS，这个数据结构可以决定哪些操作会触发VMExit进入Root Operation。进入到Root Operation后，Hypervisor取得控制权，通过读取VMCS中的VM Exit Information Fields得到发生VM Exit的原因，在vmx-

vmexit-handler函数中开始执行相应处理。Hypervisor通过选用几种调度算法（如credit、BVT等），把物理CPU合理地分配给虚拟机使用。虚拟机获得一个时间片后，在这个时间片内连续运行它的逻辑CPU，时间片消耗完后，Hypervisor会调度下一个虚拟机运行。Hypervisor正是通过VMCS结构灵活的监控虚拟机的运行，环境切换时硬件自动保存、恢复各自的状态，做到了CPU的完全隔离。

9.3.2 内存隔离

虚拟机通过内存虚拟化来实现不同虚拟机之间的内存隔离。内存虚拟化技术在客户机已有地址映射（虚拟地址和机器地址）的基础上，引入一层新的地址——“物理地址”。在虚拟化场景下，客户机OS将“虚拟地址”映射为“物理地址”；Hypervisor负责将客户机的“物理地址”映射成“机器地址”后，再交由物理处理器来执行（见图9-11）。

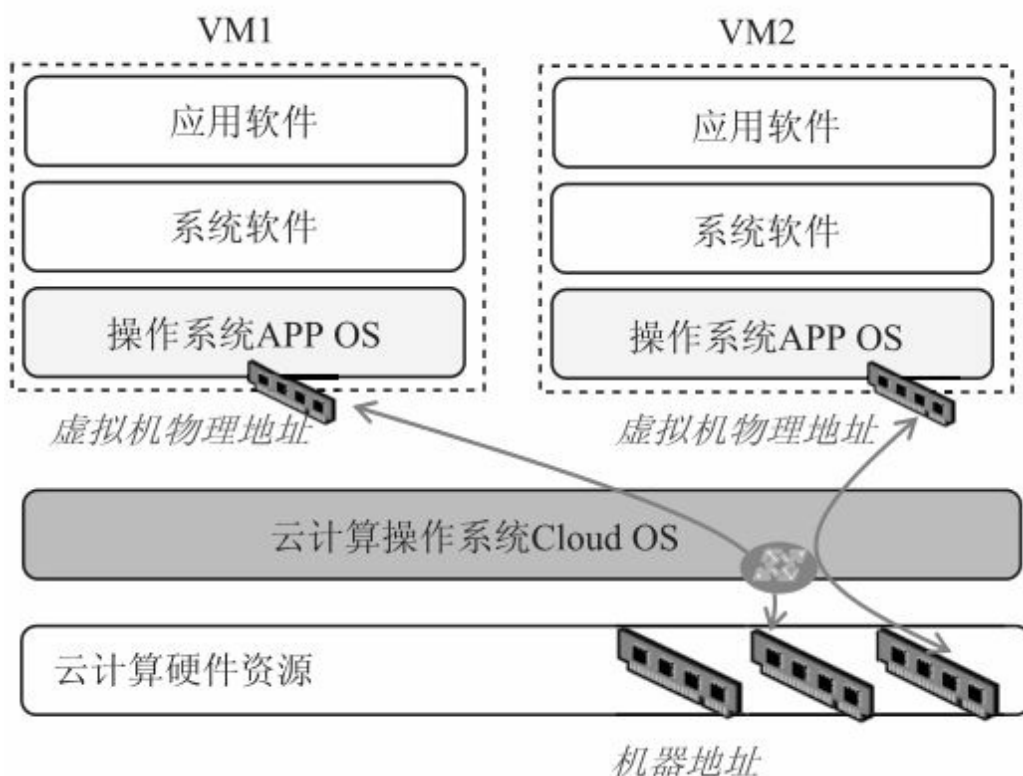


图9-11 内存隔离原理

内存虚拟化共涉及三个内存地址。

- 机器内存地址：真实的机器地址，即地址总线上出现的地址信号。
- 虚拟机物理内存地址：经Hypervisor抽象、虚拟机看到的伪物理地址。
- 应用程序内存地址：客户机OS提供给它应用程序使用的线性地址空间。

虚拟机物理内存地址与应用程序使用的内存地址之间的映射关系是由APP OS维护的，即虚拟机的OS维护虚拟机上的应用程序之间的内存分配和调度。

Cloud OS（Hypervisor）管理虚拟机使用的虚拟物理内存（Physical Memory）和真实机器内存（Machine Memory）之间的对应关系，即P2M转换，这种对应关系是一一对应的，每个虚拟机都有一张P2M表。虚拟机访问内存时，通过P2M表转换，只能访问到分配给它的内存，不能访问没有分配给它的内存，从而实现各虚拟机之间的内存隔离。

普通MMU（Memory Management Unit）只能完成一次虚拟地址到物理地址的映射，在虚拟机环境下，经过MMU转换所得到的“物理地址”并不是真正的机器地址。若需得到真正的机器地址，必须由VMM介入，再经过一次映射才能得到总线上使用的机器地址。如果虚拟机的每个内存访问都需要VMM介入，且由软件模拟地址转换的效率是很低下的，其几乎不具有实际可用性，为实现虚拟地址到机器地址的高效转换，现普遍采用的思想是：由VMM根据映射f和g生成复合的映射fg，并将这个映射关系写入MMU。

Hypervisor采用的内存硬件辅助虚拟化技术是用于替代虚拟化技术中软件实现的“影子页表”的一种硬件辅助虚拟化技术，其基本原理是：VA->PA->MA，两次地址转换都由CPU硬件自动完成（软件实现内存开销大、性能差）。Hypervisor采用VT-x技术的页表扩充技术Extended Page Table（EPT），首先VMM预先把客户机物理地址转换到机器地址的EPT页表设置到CPU中；其次客户机修改客户机页表无需VMM干预；最后，地址转换时，CPU自动查找两张页表完成客户机虚拟地址到机器地址的转换。通过使用内存的硬件辅助虚拟化技术，在客户机运行过程中无需VMM干预，去除了大量软件开销，内存访问性能接近物理机。

9.3.3 内部网络隔离

Hypervisor提供虚拟防火墙——路由器（VFR，Virtual Firewall-Router）的抽象，每个客户虚拟机都有一个或者多个在逻辑上隶属于VFR的网络接口VIF（Virtual Interface）。从一个虚拟机上发出的数据包，先到达Domain 0，由Domain 0来实现数据过滤和完整性检查，并插入和删除规则；经过认证后携带许可证，由Domain 0转发给目的虚拟机；目的虚拟机检查许可证，以决定是否接收数据包。

9.3.4 磁盘I/O隔离

虚拟机所有的I/O操作都由Hypervisor截获处理，Hypervisor保证虚拟机只能访问分配给该虚拟机的物理磁盘，实现不同虚拟机硬盘的隔离。

9.3.5 用户数据隔离

Hypervisor采用分离设备驱动模型实现I/O的虚拟化。该模型将设备驱动划分为前端驱动程序、后端驱动程序和原生驱动三个部分，其中前端驱动在DomainU中运行，后端驱动和原生驱动则在Domain0中运行。前端驱动负责将DomainU的I/O请求传递到Domain0中的后端驱动，后端驱动解析I/O请求并映射到物理设备，提交给相应的设备驱动程序控制硬件完成I/O操作。换言之，虚拟机所有的I/O操作都会由VMM截获处理；同时，系统对每个卷定义不同的访问策略，没有访问该卷权限的用户不能访问该卷，只有卷的真正使用者（或者有该卷访问权限的用户）才可以访问该卷，VMM保证虚拟机只能访问分配给它的物理磁盘空间，从而实现不同虚拟机硬盘空间的安全隔离。

9.4 虚拟化环境中的网络安全

虚拟化平台的网络通信平面划分为业务平面、存储平面和管理平面，且三个平面之间是隔离的。存储平面与业务平面、管理平面间是物理隔离；管理平面与业务平面间是逻辑隔离。通过网络平面隔离保证管理平台操作不影响业务运行，最终用户不能破坏基础平台管理。

网络平面隔离如图9-12所示。

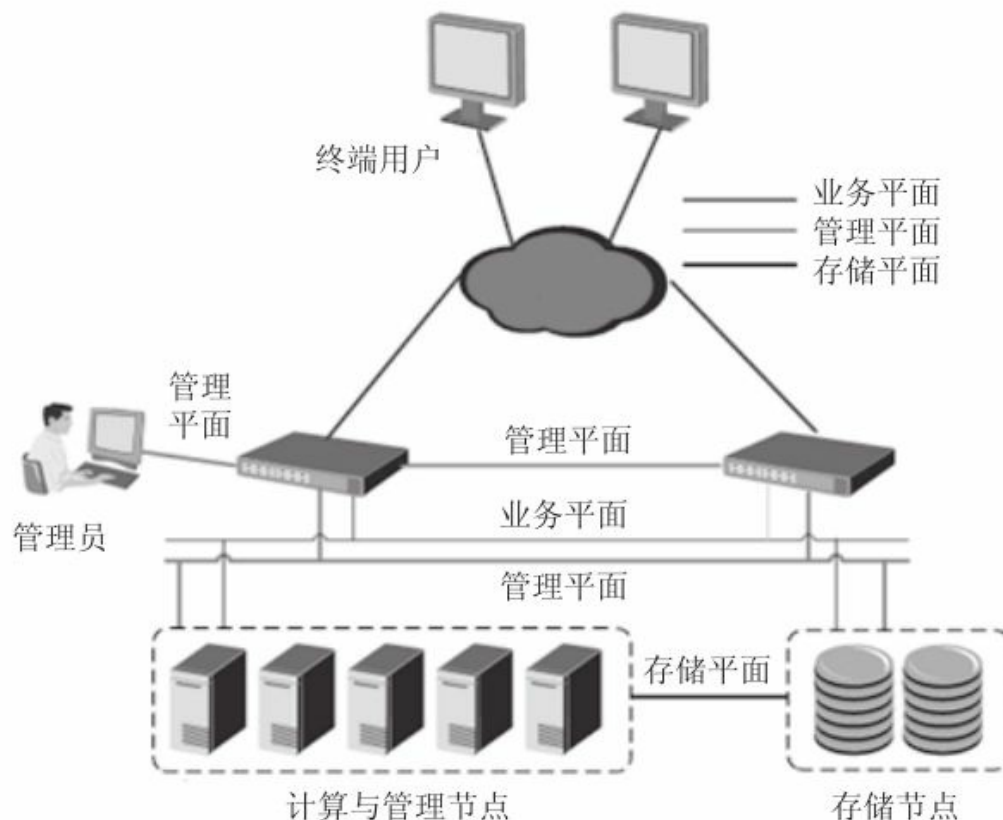


图9-12 网络平面隔离

- 业务平面：为用户提供业务通道，为虚拟机虚拟网卡的通信平面，对外提供业务应用。
- 存储平面：为iSCSI存储设备提供通信平面，并为虚拟机提供存储资源，但不直接与虚拟机通信，而通过虚拟化平台转化。
- 管理平面：负责整个云计算系统的管理、业务部署、系统加载等流量的通信。

9.4.1 虚拟交换机及防ARP攻击

ARP（Address Resolution Protocol，地址解析协议），负责将某个IP地址解析成对应的MAC地址。ARP攻击就是通过伪造IP地址和MAC地址实现ARP欺骗，通过在网络中产生大量的ARP通信量使网络阻塞，该攻击主要存在于局域网网络中，通过木马程序发起攻击。

在ARP的防攻击中，虚拟交换机EVS（Elastic Virtual Switch）起到了重要作用。那么什么是虚拟交换机呢？图9-13是EVS的架构图。

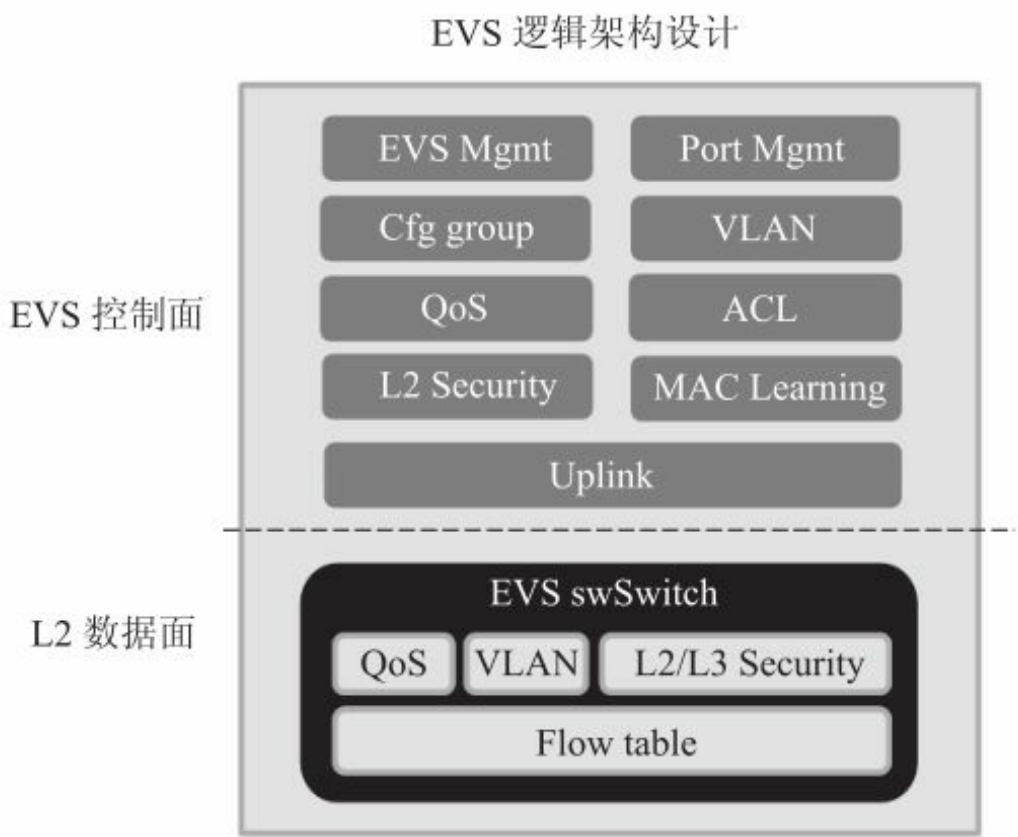


图9-13 单机EVS架构图

单机EVS功能模块职责表如表9-2所示。

表9-2 单机EVS功能模块职责表

功能模块名称	职责
EVS Mgmt	EVS对象管理，支持创建多个EVS对象
Port Mgmt	管理虚拟交换机的端口资源，根据VM申请给其分配虚拟交换机端口资源，该端口最终和VM的vNIC对应
Cfg group	提供配置组能力，方便用户对端口做批量配置
VLAN	提供标准的802.1Q能力

Qos	提供基本的Qos能力，实现对虚拟机的带宽保护、流量限速、简单的流量整形等
ACL	基于五元组，根据ACL规则对包转发引擎进行规则配置，对转发的报文根据规则进行接收/丢弃处理
L2 Security	提供二层的IP/MAC防欺骗的安全能力
Mac Learning	提供用户态的Maclearning学习能力
UpLink	管理上行链路和上行端口，连接上行链路和物理网卡，用户申请上行端口时必须连接到指定的Uplink链路上，通过VLAN能力的属性设置保证VM的流量从该上行端口（上行链路）上收发。Uplink还会管理普通nic、inic、enic等各种资源，并提供inic或者enic能力下的网络直通对应的资源分配能力
EVS.swSwitch	Host的二层虚拟交换能力，保证二层报文的跨VM、跨host的交换能力；提供QoS、VLAN、L2/L3 Security等功能，并针对每条数据流建立对应的flow table表项，后续数据流基于对应的flow table表项进行快速交换

9.4.2 IP/MAC防欺骗功能设计

弹性EVS实现的一个重要网络安全功能是IP/MAC的防欺骗功能。

其功能包括：

- 截获dhcp报文，进行解析；
- 对开启IP/MAC防欺骗功能的虚拟网卡侧过来的报文进行非法dhcp报文过滤；
- 根据开启IP/MAC防欺骗功能的虚拟网卡的dhcp ack报文生成对应IP/MAC防欺骗DB条目，用于IP报文源地址防欺骗和ARP报文防欺骗。

9.4.3 VLAN

通过虚拟网桥实现虚拟交换功能，虚拟网桥支持VLAN Tagging功能，

实现VLAN隔离，确保虚拟机之间的安全隔离。

虚拟网桥的作用是桥接一个物理机上的虚拟机实例。虚拟机的网卡eth0、eth1等称为前端接口（front-end）。后端（back-end）接口为vif，连接到Bridge。这样，虚拟机的上下行流量将直接经过Bridge转发。Bridge根据mac地址与vif接口的映射关系做数据包转发。

Bridge支持VLAN Tagging功能，这样分布在多个物理机上的同一个虚拟机安全组的虚拟机实例可以通过VLAN Tagging对数据帧进行标识。网络中的交换机和路由器可以根据VLAN标识决定是否对数据帧路由和转发，也可以依据VLAN标识提供虚拟网络的隔离功能。

如图9-14所示，处于不同物理服务器上的虚拟机通过VLAN技术可以划分在同一个局域网内，同一个服务器上的同一个VLAN内的虚拟机之间通过虚拟交换机进行通信，而不同服务器上的同一VLAN内的虚拟机之间通过交换机进行通信，确保不同局域网的虚拟机之间的网络是隔离的，不能进行数据交换。

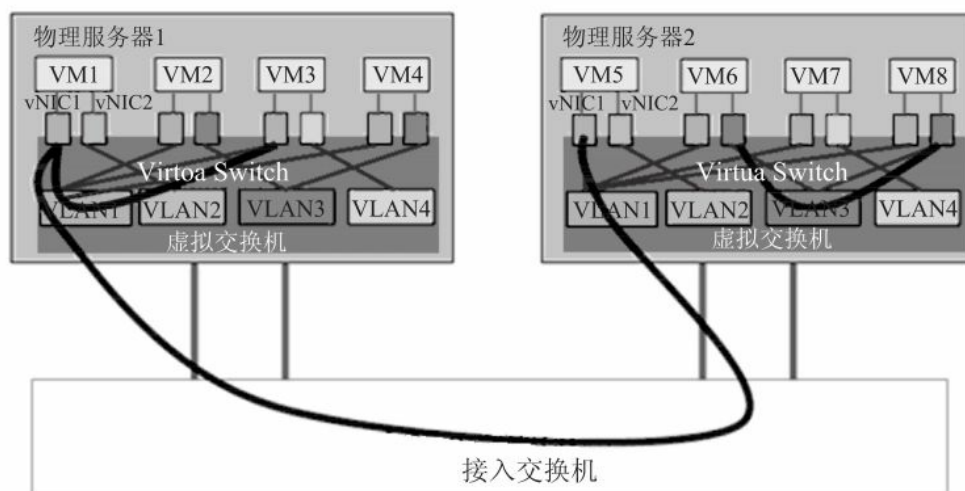


图9-14 VLAN组网图

9.5 云数据安全

数据是云计算的核心，数据的安全包括数据的机密性保护（张三的数据不能被李四读取）、完整性保护（数据不能被篡改）、数据操作审计等内容。下面介绍云数据安全的一些解决方案。

9.5.1 云存储加密与用户数据安全

云计算虚拟磁盘加密方案

云计算用户最大的担心就是个人数据安全，由于云计算将数据集中管理，数据不再由用户自己管控，而是被云计算运营企业、云计算管理员控制，如何保证这些用户的数据不被偷看、泄露？如何保证用户数据不会在不同用户间交叉，造成泄露？数据加密是其中最可信的解决方案，从密码理论上讲，只要用户的密钥没暴露，即使数据丢失也能保障信息不外泄（见图9-15）。

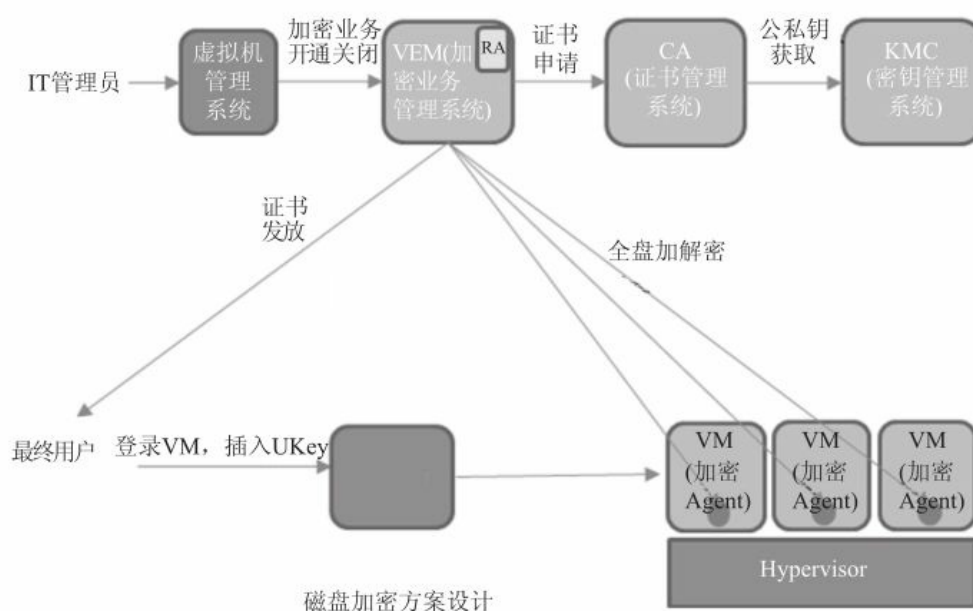


图9-15 云计算虚拟磁盘加密系统（VES）

云计算数据安全解决方案

对虚拟机数据盘进行全盘加密，加密过程通过降低CPU消耗来避免影响系统效率。采用加密机、PKI证书、对称密钥三级密钥机制提供对用户虚拟磁盘的高强度、高安全性加密。数据安全防护系统与第三方CA无缝对接，提供高安全、高可靠的密钥管理服务。通过屏蔽云平台差异，兼容所有类型的Hypervisor，实现与云业务管理系统松耦合关系以及数据安全防护系统的独立部署。

9.5.2 用户数据安全有效保护

数据安全是保障数据中心安全的重点。为了保障用户的数据安全，需要从数据隔离、访问控制等多个方面采取措施。

用户卷访问控制

系统对每个卷定义不同的访问策略，没有访问该卷权限的用户不能访问该卷，只有卷的真正使用者（或者有该卷的访问权限）才可以访问该卷，每个卷之间是互相隔离的。

存储节点接入认证

➤ 存储节点采用标准的iSCSI进行访问，并且支持询问握手认证协议（CHAP, Challenge Handshake Authentication Protocol）认证功能。CHAP协议可通过三次握手，周期性校验对端的身份。CHAP认证可在初始链路建立时、完成时以及在链路建立之后重复进行。通过递增改变的标识符和可变的询问值，可防止来自端点的重放攻击，限制暴露于单个攻击的时间。CHAP认证功能可以提高应用服务器访问存储系统的安全性。

➤ 存储系统启用CHAP认证以后，应用服务器侧也必须启用CHAP认证，同时在存储系统中把应用服务器的信息加入到存储系统的合法CHAP用户，只有经过CHAP认证通过以后，才能连接到存储系统并存取数据。

剩余数据删除

➤ 当用户把卷卸载释放后，系统在把该卷进行重新分配之前，会对该卷进行数据格式化，以保证该卷上的用户数据的安全性。

➤ 存储的用户文件/对象删除后，对应的存储区进行完整的数据擦除，并标识为只写（只能被新的数据覆写），保证不被非法恢复。

数据备份

➤ 云数据中心的数据存储采用多重备份机制，每一份数据都可以有一个或者多个备份，当数据因存储载体（如硬盘）出现故障的时候，不会引起数据的丢失，也不会影响系统的正常使用。

➤ 系统对存储数据按位或字节的方式进行数据校验，并把校验的信息均匀地分散到阵列的各个磁盘上。阵列的磁盘上既有数据，也有数据校验信息，数据块和对应的校验信息会存储于不同的磁盘上，当一个数据盘损坏时，系统可以根据同一带区的其他数据块和对应的校验信息来重构损坏的数据。

IPSAN保险箱技术

存储系统遭遇意外掉电时，采用数据保险箱技术保证数据的安全性和完整性。数据保险箱技术即从系统中的某几块硬盘上划分出一定区域，用来专门存放因突然掉电而尚未及时写入硬盘的Cache数据和一些系统配置信息。当系统外部供电中断时，则通过内置电池或外置UPS供电，使得Cache中的数据能够写入数据保险箱。当外部电力恢复时，控制器再将数据从数据保险箱中读回到Cache，继续完成对数据的处理。

9.6 公有云、私有云的安全组

虚拟化带来的最大威胁是虚拟机间资源未完全隔离，云计算提供了同一物理机上不同虚拟机之间的资源隔离，避免虚拟机之间的数据窃取或恶意攻击，保证虚拟机的资源使用不受周边虚拟机的影响。终端用户使用虚拟机时，仅能访问属于自己的虚拟机的资源（如硬件、软件和数据），不能访问其他虚拟机的资源，保证虚拟机隔离安全。

实现原理如图9-16所示。

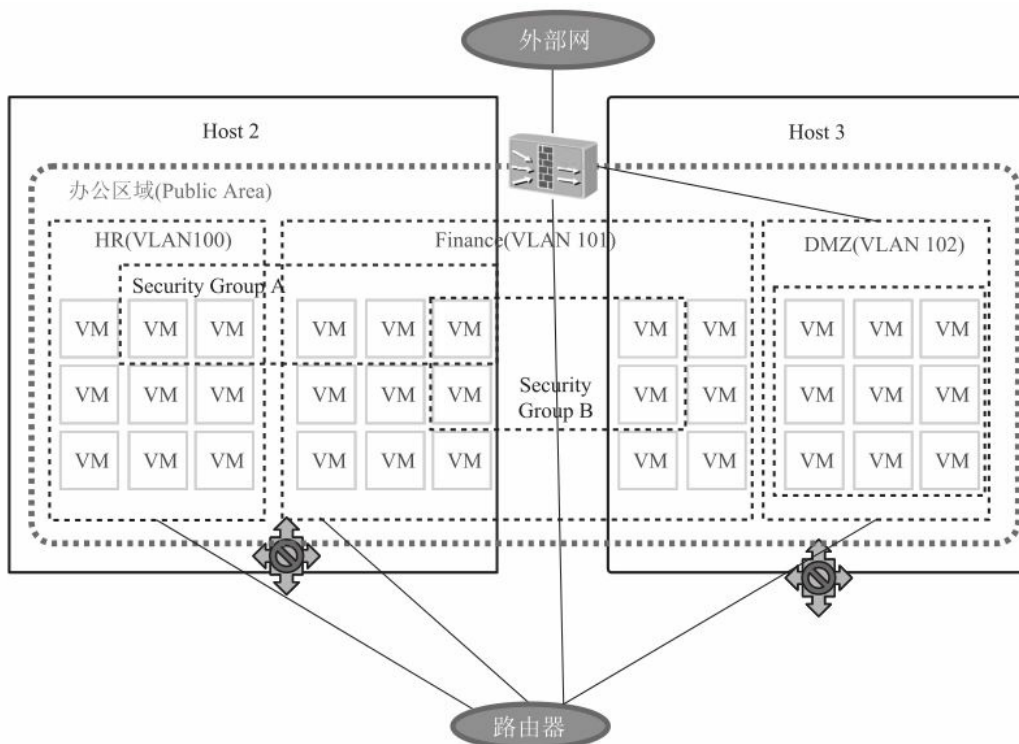


图9-16 安全组原理

用户可以根据虚拟机的属性灵活定义安全组，包括IP、MAC、VLAN、Subnet等元数据。安全组是具有同等安全要求的一组虚拟机。安全组可以由一个或多个VM，也可以由一个或多个VLAN组成。安全组的策略可以细化到VLAN甚至每台VM。每个安全组对应一个安全域，可以基于安全组定义自己的访问控制规则，实现域内和域外的安全隔离。安全组间的访问控制在Hypervisor层实现，每个host上部署一个，不占用额外的虚拟机资源，不存在安全检测引流导致的流量迂回问题。

安全网关控制器（Security Gateway Controller, SGC）是实现VM隔离功能的核心组件，它部署在Hypervisor，主要功能如下：

- 当SGC上不存在从源VM到目的VM的安全组规则时，它和目的端SGC进行安全组规则协商，以决定是否允许VM之间的通信；
- 当SGC本地的安全组成员表、安全组规则表发生变化时，需要通知相关SGC做相应的更新；
- SGC接收安全组规则协商请求消息，根据本地的安全组成员表

和安全组规则表，动态生成安全组会话表；

➤ SGC接收安全组成员、安全组规则表（增、删、改）通知消息，以及VM迁移事件，并做相应处理；

➤ SGC接收Hypervisor发来的VM在线、离线事件，并做相应处理。

9.7 云安全管理

9.7.1 日志管理

云计算带来了成本降低、效率提高等一系列好处的同时，由于计算、存储的集中，对管理维护提出了更高的安全要求，以保障基础设施的安全运行。

系统支持集中的日志收集和存储，满足各种审计要求，如分级保护。

一般的云计算平台均支持以下三类日志。

（1）操作日志

操作日志记录操作维护人员在管理节点进行的管理维护操作，包括用户、操作类型、客户端IP、关键参数、操作时间、操作结果等内容，存放在管理节点的数据库中。审计人员可通过OMS Portal导出和查看操作日志，定期审计操作维护人员在管理节点进行操作，及时发现不当或恶意的操作。操作日志也可作为抗抵赖的证据。

（2）运行日志

运行日志记录各节点的运行情况，分为debug、info、warning、error 4个级别，优先级依次递增，可由日志级别来控制日志的输出。

各节点的运行日志通过日志管理组件统一汇总、过滤成高级别日志（warning、error级别）和完整日志（包括所有已设置输出级别的日志包）。高级别日志定期汇总到日志服务器统一存放。完整日志在本地存放，并支持通过脚本方式上载指定节点、时间段的完整日志到日志服务

器。

运行日志包括级别、线程名称、运行信息等内容，维护人员可通过查看运行日志，了解和分析系统的运行状况，及时发现和处理异常情况。

（3）黑匣子日志

黑匣子日志记录系统严重故障时的定位信息，主要用于故障定位和故障处理，便于快速恢复业务。其中计算节点产生的黑匣子日志汇总到日志服务器统一存放，而管理节点、存储节点产生的黑匣子日志在本地存放。

云计算系统支持集中的日志收集和存储，如图9-17所示。

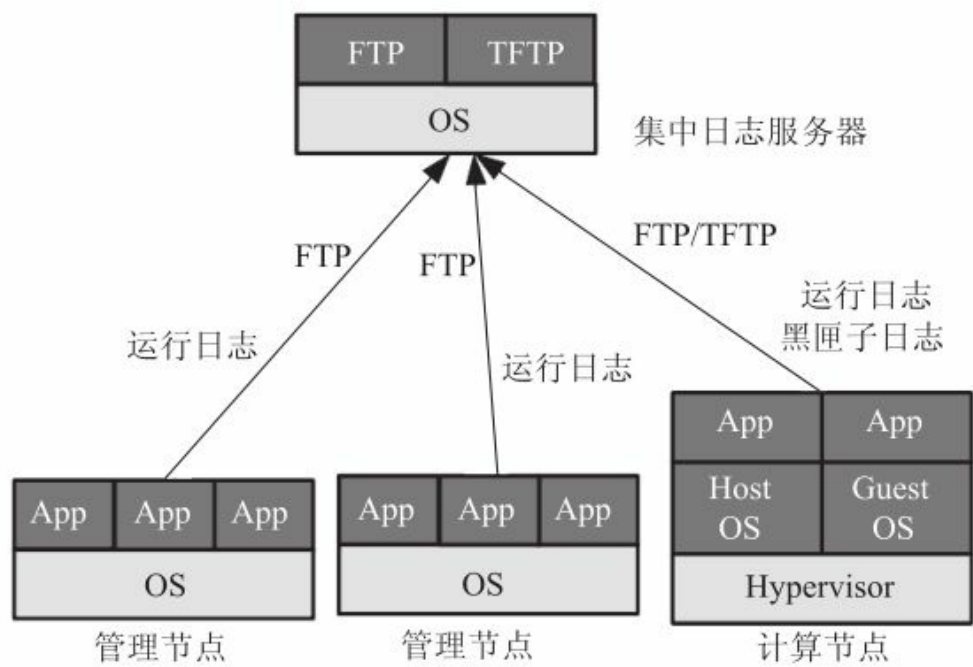


图9-17 集中日志管理

在各节点部署日志收集客户端，实时收集本地产生的运行日志、黑匣子日志，通过配置日志收集服务器，实现将日志数据过滤成高级别日志和完整日志。高级别日志定期汇总到集中日志服务器。完整日志通常存在本地节点，可通过脚本把指定节点、指定时间段的完整日志汇总到日志服务器进行集中管理，主要管理方法如下。

安全告警管理

安全告警是指当系统侦测到违背安全策略的事件行为时，将与安全事件相关的一些信息上报给安全告警管理，管理员根据这些信息对违背安全策略的行为进行及时处理，排除安全隐患。安全告警上报的内容包含告警的来源、告警产生的时间、告警产生的原因、服务提供者、服务使用者、告警级别、事件类型等信息。

日志分类管理

通过查看日志可以了解系统的运行情况和操作记录，用于用户行为审计和问题定位。日志分为操作日志和运行日志，操作日志与系统安全相关。操作日志记录了用户对系统所做的操作以及操作的结果，用于跟踪和审计。记录操作日志，可以快速定位系统是否遭受恶意的操作和攻击。

日志备份

系统需要定期备份操作日志，并提供“定时周期性备份方式”备份操作日志；日志备份成功后，系统会自动删除已备份的日志。

在界面上查看日志时，采用Https方式进行传输。查看日志时采取的安全措施如下。

- 根据管理员的权限定义日志查看范围。超级管理员能查看所有日志，但普通管理员仅能查看自己的操作日志。任何人员不能在界面上修改或删除日志。有查询权限的人才能导出日志。
- 日志格式：各系统针对操作日志提供多个字段，详细记录外界用户在系统上执行的具体操作、用户信息、操作时间等信息，便于管理员根据操作日志识别有风险的操作或已导致问题的危险操作，并采取合理的防卫措施，提高系统的安全性。

9.7.2 账户和密码安全

硬件设备及系统存在初始默认的账号和密码，主要是用于维护硬件设备及系统。建议管理员首次登录就修改默认密码，且修改时需满足密码复杂度要求，同时建议管理员定期修改密码，确保密码不泄漏。

系统中各类账户的密码加密、设置和修改原则如表9-3所示。

表9-3 各类账户的密码加密、设置和修改原则

项	原则
目	
首次设置原则	密码 首次登录系统时，需要修改系统默认密码，密码修改方法请参见账户密码维护；密码设置需满足密码策略表要求
加密原则	密码 所有密码加密存放；密码不以明文形式在界面显示
修改原则	密码 只有通过认证的用户，才能修改密码；修改密码时，必须通过旧密码校验；密码达到最长有效期后，用户再次登录时，系统强制用户更改密码；密码需要定期修改，对于管理员密码，建议最长180天修改一次
策略表修改原则	密码 系统安装时生成默认的密码策略表；密码策略表只有系统管理员有权修改；修改密码策略表后，根据原有密码策略表设置的密码能够正常登录

9.7.3 分权分域管理

通过对管理员区分权限，对被访问的数据区分权限，限制管理员访问系统的范围，保证系统的安全。管理员分权分域管理模型遵循美国国家标准与技术研究院（NIST, The National Institute of Standards and Technology）标准的基于角色的访问控制（RBAC, Role Based Access Control）模型。

三权分立

底层采用强制访问控制措施，在系统启动的时候就自动产生系统管理员、审计员和操作员三类角色，而且每类角色可以延伸新的角色，系统

管理员不能删除其他角色，其行为将会被审计员所审计。

分权

“分权”指区分“操作权限”，它由“角色”进行控制。一个“角色”可拥有一个或多个不同的“操作权限”，一个“用户”可拥有一个或多个不同的“角色”。通过绑定“用户”和“角色”，实现“用户”和“操作权限”的绑定。如果一个“用户”拥有多个“角色”，其拥有的“操作权限”是多个“角色”拥有的“操作权限”的并集。产品支持灵活的设置角色，并灵活赋予角色拥有的权限。

分域

“分域”指区分“数据管理范围”。

基础设施及虚拟桌面分权分域管理

创建多个管理员用户，并支持对各管理员用户进行操作权限和数据权限的管理。“超级管理员”可根据企业自身的业务需要，在“桌面云业务维护系统”系统中添加“业务管理员”，通过“业务管理员”管理和维护企业的桌面云资源。“超级管理员”可赋予“业务管理员”不同的角色，用以定义用户可执行的操作权限。“超级管理员”可配置“业务管理员”，用以定义用户可执行操作的范围。这里提到的“超级管理员”不一定是一个人，出于安全需要，可能是以多个人联合完成的，比如5个人，每个人拥有一段独立的密码，5个人分别输入密码才能让超级管理员权限生效，而超级管理员的操作则在5个人的监督下完成。

9.8 云安全应用实施案例

某单位以前采用传统PC办公，终端数量众多，维护工作量大，分级保护终端软件安装和维护成本巨大，使用和维护极其不便；即使在此情况下，泄密风险也时刻存在。因此应重视桌面云技术解决安全问题，实现安全和方便地办公。

云桌面管理包括桌面管理和虚拟化管理两个部分。虚拟桌面管理软件提供高性能且可靠的桌面投送。虚拟化管理对硬件设备（服务器、存储、交换机）、虚拟资源进行集中管理，采用B/S架构，可以远程统一管理

本项目中VDI桌面、应用虚拟化、服务器虚拟化三个资源池。虚拟化管理系统可管理、监控硬件资源、虚拟资源，支持虚拟机的快速部署、定制化策略调度。

桌面云登录认证采用身份认证网关和AD认证共存的解决方案，对最终用户体现为USBkey登录认证，桌面云内部实际认证利用AD的域用户名/密码，由身份认证网关/管控平台实现用户证书与用户名/密码的映射转换。

身份认证网关部署在安全接入网关前，登录WI的会话请求被网关客户端截获，送到身份认证网关（由网关客户端与身份认证网关建立安全加密通道），经过解析处理（如域用户密码代填）后转发到WI（见图9-18），可登录WI（网页界面，Web Interface）或通过虚拟机实现单点登录（只输入一次PIN码）。

管理平台

- 接受管控平台客户端的USBkey注册。
- USBkey认证的用户管理，包括用户与证书的对应关系，以及维护证书与用户名/密码的映射，供认证网关通过登录模块查询域用户名和密码。
- 定期自动修改用户的域密码。

认证网关

- 用户USBkey的登录认证：认证通过后代填用户名/密码，然后登录到WI。

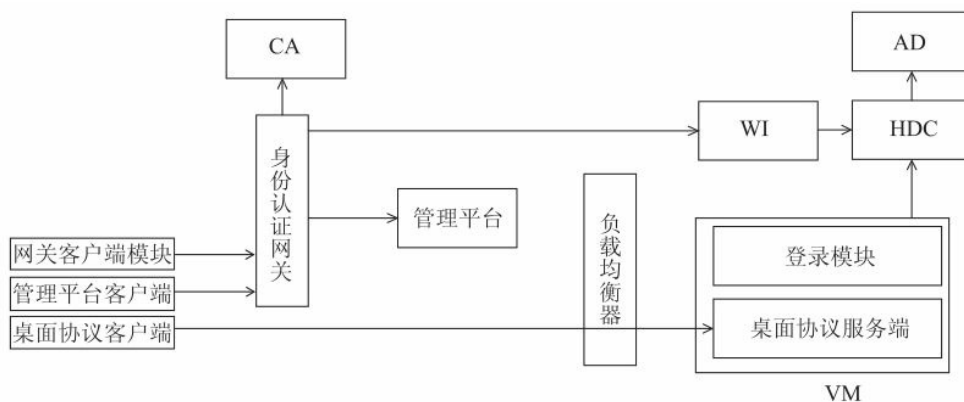


图9-18 桌面云系统安全架构

- 网关客户端模块：拦截TC到Wi的登录请求，代理转发到认证网关。
- 管控平台客户端：USBkey使用前注册到管控平台，第一次注册需要输入用户的域用户名和密码；截获PIN码到管控平台中（为保证PIN码、密码的安全，需经过认证网关）。
- 登录模块：从管控平台获取PIN码、用户名和密码，然后登录虚拟机。

9.9 云计算安全的其他考虑

安全与用户体验

云计算体系采用的安全措施越多，该云体系的投入和运营成本相对没有安全措施云计算体系的投入与成本会越高，出于安全的需要，运营维护效率也会相对降低。举个最简单的例子，大家在使用互联网时最头疼的一个问题就是密码，不同的网站对密码的格式要求不同，导致我们需要记下很多用户名密码，密码忘记是常事，找回密码往往还需另外一个密码。提升体系安全性的同时，怎样改善用户体验，还需要持续不断的改进。利用桌面云安全方案来替换以前的PC安全方案便是一个很好的例子。桌面云安全性不仅比PC时代提升了，而且终端用户体验以及运维管理效率还能得到大幅提升，这无疑是涉密企业与机构的极佳选择。

安全与效益

作为企业而言，要综合地考虑安全问题，因为企业需要效率和赢利。比如，对于当今已经竞争白热化的企业电子商务，安全问题很重要，因为每个安全漏洞可能都意味着企业的损失，但出于竞争的需要，企业需要不断创新交易模式，比如交易模式由PC的互联网点对点固定交易过渡到手机的移动交易模式，以此来抢占更多的市场空间。创新本身就会有安全风险（比如手机更容易丢失，更容易被植入恶意软件），企业若想抢占先机，必须去承担这个风险，不能因此退缩不前。被竞争对手超越所带来的损失，可能比黑客窃取一部分资金的损失大得多。可以由此衍生出保险模式，比如一些企业以“你敢付、我就敢赔”这种承担安全风险的前卫模式，大力拓展其市场空间。

第10章 大数据平台核心技术与架构

10.1 大数据特点与支撑技术

从技术角度来看，大数据所涵盖的范围往往比人们想象的窄很多。因为无论是大数据的存取，还是大数据的处理，大数据的分析要受限于当代IT技术发展的制约。正因为IT技术的制约，传统IT技术存储处理能力受限，所以从IT技术角度才感觉数据太“大”了。这个“大”表现在数据量的大，比如大部分存储设备都是TB级，几十TB到数百TB以及对传统存储而言是很高的数据量了。那么几十PB、成百上千PB的数据量自然对当前存储系统而言是“大数据”了。存储数据量的“大”很有时效性，因为大家都清楚，20世纪90年代初，1GB的存储空间都觉得“大”得不得了。现在说1GB，连PC机内存配置都觉得不够。数据量级之外的大数据，涵指数据格式的多样性，特别是大家已经熟知的图片、音频、视频、网页、日志等半结构化、非结构化的数据。除数据格式多样性之外，其还有数据实时性要求、数据价值密度等维度的指代。这就是通常大家比较认可的大数据4V特性。

- 体量（Volume）：非结构化数据超大规模地增长，占总数据量的80%~90%，比结构化数据增长快10倍到50倍，是传统数据仓库的10倍到50倍。
- 多样性（Variety）：大数据具有异构的多样性，拥有许多不同形式（文本、图像、视频、机器数据），无模式或者模式不明显，拥有不连贯的语法或句义。
- 价值密度（Value）：有大量的不相关信息，对未来趋势与模式可预测分析，可进行深度复杂分析（机器学习、人工智能等）。
- 速度（Velocity）：实时分析而非批量式分析，对于实时的数据输入、处理与丢弃，分析结果立竿见影而非事后见效。

首先是对Volume的理解，非结构化数据一直存在，随着处理成本的降低和竞争的加剧，非结构化数据受到越来越多的重视。非结构化数据并

不是完全没有“结构”的数据，而是包括类似图像、视频等比较难以计算的数据，以及日志等“结构”变化相对随意、不严格受控的半结构化数据。

第二，无论是结构化大数据，还是非结构化数据，处理的挑战都非常大，由于竞争的加剧，商业过程要求从这些数据中进行提取有效信息的时间大大缩短。数据量变大的同时，还要求提取过程缩短，并且提取过程本身也越来越复杂化，甚至对同一个业务问题的解答，要求用不同模型或者同一模型的不同参数加以对比印证。

不同性质、不同来源但是与同一个商业对象（比如客户）相关的各种数据（比如流数据、块数据和全局数据）都必须在一个系统中进行有效整合，形成对这个商业对象的有效认知，从而驱动商业流程。

大数据环境下，数据来源非常丰富且数据类型多样，存储和分析挖掘的数据量庞大，对数据展现的要求较高，并且很看重数据处理的高效性和可用性。传统的数据采集来源单一，且存储、管理和分析数据量也相对较小，大多采用关系型数据库和并行数据仓库即可处理。对依靠并行计算提升数据处理速度方面而言，传统的并行数据库技术追求事务写的一致性和容错性，难以保证其可用性和扩展性。传统的数据处理方法是以处理器为中心，而大数据环境下，需要采取以数据为中心的模式，减少数据移动带来的开销。因此，传统的数据处理方法已经不能适应大数据的需求。针对非结构化数据，利用分布式文件系统和MapReduce技术进行处理的Hadoop技术；针对实时类数据，采用流处理技术，如S4、Esper、Storm技术；针对结构化数据，采用已相对比较成熟的关系型数据库技术，如MPP RDB技术、SMP OLTP、MPP OLAP技术等。但是，任何技术的发展都是以业务为根本驱动的（见图10-1）。

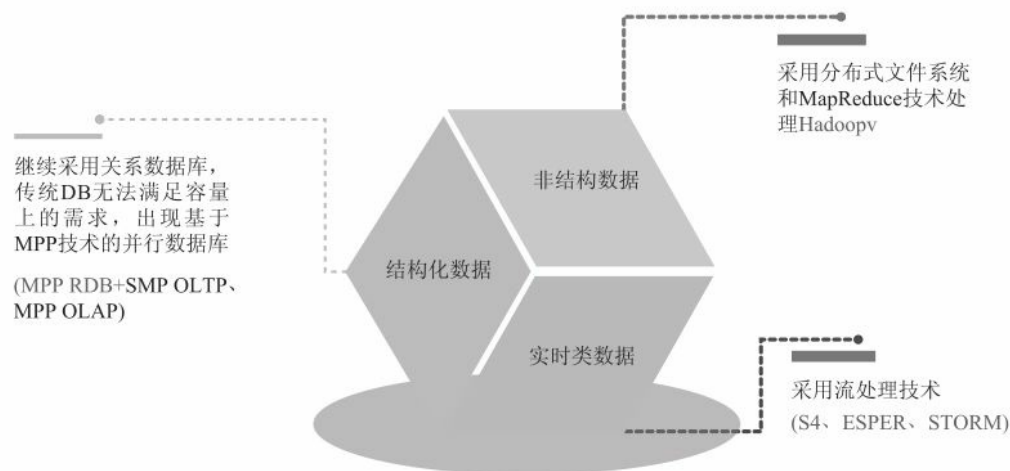


图10-1 大数据处理技术

10.1.1 数据采集技术

数据是指通过传感器网络、社交网络及移动互联网等方式获得的各种类型的结构化、半结构化及非结构化的海量数据。要重点突破分布式高速、高可靠数据的抽取或采集、高速数据全映像等大数据收集技术；突破高速数据解析、转换与装载等大数据整合技术；设计质量评估模型，开发数据质量技术。

大数据采集一般分为大数据智能感知层：主要包括数据传感体系、网络通信体系、传感适配体系、智能识别体系及软硬件资源接入系统，实现对结构化、半结构化、非结构化的海量数据的智能化识别、定位、跟踪、接入、传输、信号转换、监控、初步处理和管理等。必须着重攻克针对大数据源的智能识别、感知、适配、传输、接入等技术。重点攻克分布式虚拟存储技术，大数据获取、存储、组织、分析和决策操作的可视化接口技术，大数据的网络传输与压缩技术，大数据隐私保护技术，等等。

10.1.2 数据预处理技术

本环节主要完成对已接收数据的辨析、抽取、清洗等操作。因获取的数据可能具有多种结构和类型，数据抽取过程可以帮助我们将这些复杂的数据转化为单一的或者便于处理的类型，以达到快速分析处理的目的。所有的大数据并不全是有价值的，有些数据并不是我们所关心的内容，而另一些数据则是完全错误的干扰项，因此要对数据进行过滤“去噪”，

从而提取出有效的数据。

10.1.3 数据存储及管理技术

数据存储与管理要用存储器把采集到的数据存储起来，建立相应的数据库，并进行管理和调用。其重点解决复杂结构化、半结构化和非结构化大数据管理与处理技术，包括大数据的可存储、可表示、可处理、可靠性及有效传输等几个关键问题。大数据存储与管理还需要可靠的分布式文件系统、能效优化的存储、计算融入存储、大数据的去冗余及高效低成本的大数据存储技术、分布式非关系型大数据管理与处理技术、异构数据的数据融合技术，数据组织技术、大数据建模技术、大数据索引技术、大数据移动/备份/复制等技术、开发大数据可视化技术等。

10.1.4 数据分析及挖掘技术

大数据分析技术最近几年获得了很大的进展，包括改进已有数据挖掘算法和机器学习技术；开发数据网络挖掘、特异群组挖掘、图挖掘等新型数据挖掘技术；突破基于对象的数据连接、相似性连接等大数据融合技术；突破用户兴趣分析、网络行为分析、情感语义分析等面向领域的大数据挖掘技术。

数据挖掘就是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中的、人们事先不知道的但又是潜在有用的信息和知识的过程。数据挖掘涉及的技术方法很多，有多种分类方法。

根据挖掘任务可分为分类或预测模型发现、数据总结、聚类、关联规则发现、序列模式发现、依赖关系或依赖模型发现、异常和趋势发现等。

根据挖掘对象可分为关系数据库、面向对象数据库、空间数据库、时态数据库、文本数据源、多媒体数据库、异质数据库、遗产数据库以及环球网Web等。

根据挖掘方法分，可粗分为机器学习方法、统计方法、神经网络方法和数据库方法。机器学习中，可细分为归纳学习方法（决策树、规则归纳等）、基于范例学习、遗传算法等。统计方法中，可细分为回归分析（多元回归、自回归等）、判别分析（贝叶斯判别、费歇尔判别、非参数判别等）、聚类分析（系统聚类、动态聚类等）、探索性分析（主元分析法、相关分析法等）等。神经网络方法中，可细分为前向神经网络

（BP算法等）、自组织神经网络（自组织特征映射、竞争学习等）等。数据库方法主要是多维数据分析或OLAP方法，另外还有面向属性的归纳方法。

我们可从挖掘任务和挖掘方法的角度，着重突破。

- 可视化分析。数据可视化无论对于普通用户或是数据分析专家，都是最基本的功能。数据图像化可以让数据自己说话，让用户直观地感受到结果。
- 数据挖掘算法。图像化是将机器语言翻译给人看，而数据挖掘就是机器的母语。分割、聚类、孤立点分析还有五花八门的算法供我们精炼数据，挖掘价值。这些算法一定要能够应付大数据的量，同时具有很高的处理速度。
- 预测性分析。预测性分析可以让分析师根据图像化分析和数据挖掘的结果做出一些前瞻性判断。
- 语义引擎。语义引擎需要有足够的人工智能以从数据中主动地提取信息。语言处理技术包括机器翻译、情感分析、舆情分析、智能输入、问答系统等。
- 数据质量和数据管理。数据质量与管理是管理的最佳实践，透过标准化流程和机器对数据进行处理可以确保获得一个预设质量的分析结果。

10.1.5 数据展现与应用技术

大数据技术能够将隐藏于海量数据中的信息和知识挖掘出来，为人类的社会经济活动提供依据，从而提高各个领域的运行效率，大大提高整个社会经济的集约化程度。在我国，大数据将重点应用于以下几大领域：金融、电商、公共服务，例如商业智能技术、政府决策技术、电信数据信息处理与挖掘技术、电网数据信息处理与挖掘技术、气象信息分析技术、环境监测技术、大规模基因序列分析比对技术、Web信息挖掘技术、多媒体数据并行化处理技术、影视制作渲染技术以及其他各种行业的云计算和海量数据处理应用技术等。

10.2 企业级Hadoop

10.2.1 Apache Hadoop起源

Hadoop由Apache基金会于2005年秋天作为Lucene的子项目Nutch的一部分正式引入，该技术受到最先由Google开发的Map/Reduce和Google File System（GFS）的启发。2006年3月，Map/Reduce和HDFS（GFS的开源实现）分别被纳入称为Hadoop的项目中。Facebook向Apache基金会贡献了Hive后，Hadoop系统具备了以类SQL方式处理结构化数据的能力。2010年9月，Hive脱离Hadoop，成为Apache顶级项目。HDFS、MapReduce、Hive分别对应分布式系统的存储、计算框架、结构化分析能力。

MapReduce是Google在2004年提出的一个编程模型，用于大规模数据集（大于1TB）的并行运算。概念“Map（映射）”、“Reduce（化简）”以及其主要思想都是从函数式编程语言里借来的，并从矢量编程语言里借来了特性。其极大地方便了编程人员在不会分布式并行编程的情况下，将自己的程序运行在分布式系统上。

GFS（Google File System）是Google在2003年发表的文章，但现在仍被广泛讨论，其对后来的分布式文件系统设计具有指导意义。GFS的主要假设为GFS的服务器都是普通的商用计算机，并不可靠，集群出现节点故障是常态。系统存储适当数量的大文件，理想的负载是几百万个文件，文件一般都超过100MB，GB级别以上的文件是很常见的，必须进行有效管理。负载通常包含两种读，即大型的流式读（顺序读）和小型的随机读。前者通常一次读数百KB以上，后者通常在随机位置读几个KB。从这些假设基本可以看出GFS期望的应用场景应该是大文件、连续读、不修改、高并发。HDFS是GFS的开源实现。

Hive是Hadoop项目中的一个子项目，由Facebook向Apache基金会贡献。Hive被视为一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并可以将SQL语句转换为MapReduce任务进行运行。其优点是学习成本低，可以通过类SQL语句快速实现简单的MapReduce任务，不必开发专门的MapReduce应用，十分适合数据仓库的统计分析。

从方案的架构来看，Hadoop已形成一整套完整的生态环境，是当前最重要的大数据平台之一，具有良好的并行处理能力、可扩展性和伸缩能

力，非常适合处理半结构化、非结构化的类文本数据，如网页、日志等。

10.2.2 企业级Hadoop总体框架

企业级Hadoop可系统化地对Apache Hadoop进行增强，让对Hadoop缺乏了解的专业人员或者只有少量Hadoop专业人员的企业能够利用Hadoop的处理能力，解决本企业面临的与大数据相关业务挑战。

企业级Hadoop对外提供大容量数据分析和查询能力，解决各大企业的以下需求（见图10-2）：

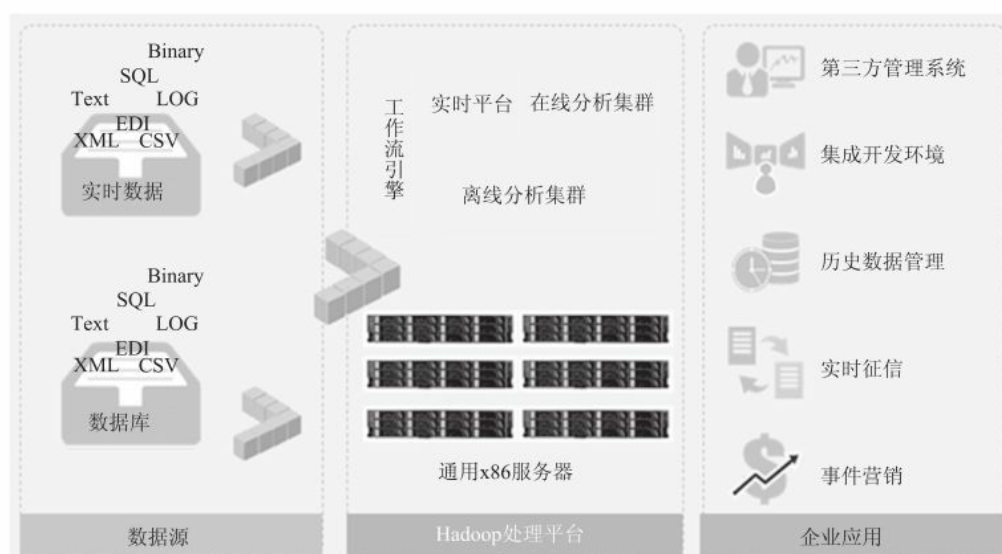


图10-2 Hadoop企业产品定位

- 快速地整合和管理不同类型的大容量数据；
- 对原生形式的信息采用高级分析；
- 可视化所有的可用数据，供特殊分析使用；
- 为构建新的分析应用程序提供开发环境；
- 工作负荷的优化和调度。

产品化的Hadoop需要在开源Hadoop版本的基础上对HBase、HDFS和MapReduce等组件增加HA、查询和分析功能，并进行性能优化。

Hadoop需要支持多种服务器的硬件平台，供企业根据自身需求灵活选择。用户可以通过简单地叠加相应服务的内存要求来计算所需要的内存总和。

产品化Hadoop软件系统整体结构如图10-3所示。

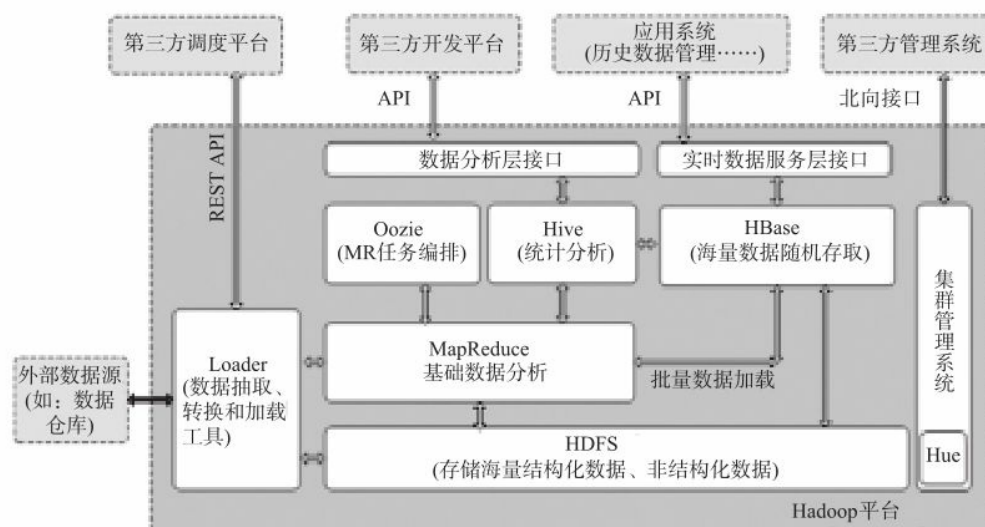


图10-3 Hadoop软件系统架构

企业级Hadoop产品，需要对开源组件进行封装和增强，对外提供稳定的数据分布式存储和分析能力，包括数据的访问、存储、处理和保护功能，分为HDFS、HBase、MapReduce和ZooKeeper。

- **HDFS:** Hadoop分布式文件系统（Hadoop Distributed File System）能提供高吞吐量的数据访问，适合大规模数据集方面的应用。
- **HBase:** 提供海量数据存储功能，是一种构建在HDFS之上的分布式、面向列的存储系统。
- **MapReduce:** 提供快速并行处理大量数据的能力，是一种分布式数据处理模式和执行环境。

➤ **ZooKeeper:** 提供分布式、高可用性的协调服务能力，帮助系统避免单点故障，从而建立可靠的应用程序。

10.2.3 HDFS

HDFS，即Hadoop分布式文件系统（Hadoop Distributed File System），能提供高吞吐量的数据访问，适合大规模数据集方面的应用。HDFS包含主、备NameNode和多个DataNode。HDFS是一个Master/Slave的结构，在Master上运行NameNode，而在每一个Slave上运行DataNode。NameNode和DataNode之间的通信都是建立在TCP/IP的基础之上的。NameNode和DataNode均被设计为可以部署在Linux服务器上（见图10-4、表10-1）。

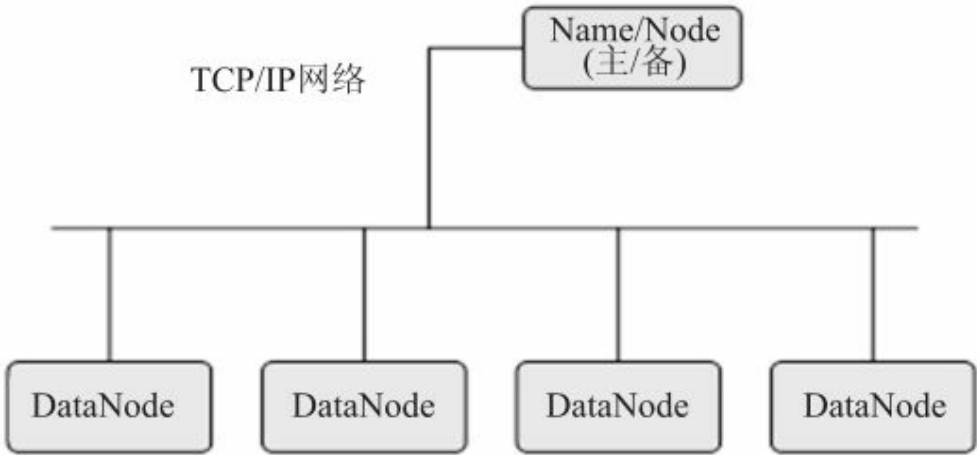


图10-4 HDFS结构

表10-1 HDFS各功能模块说明表

名称	描述
NameNode	用于管理文件系统的命名空间、目录结构、元数据信息以及提供备份机制等，分为： ➤ Active NameNode: 管理文件系统的命名空间、维护文件系统的目录结构树以及NameNode元数据信息，记录写入的每个“数据块”与其归属文件的对应关系

➤ **Secondary NameNode:** 对Active NameNode进行监控，对Active NameNode中的数据进行备份，随时准备在Active NameNode出现异常时接管其服务

DataNode 用于存储每个文件的“数据块”数据，并且会周期性地向NameNode报告该DataNode的存放情况

HDFS原理如图10-5所示。

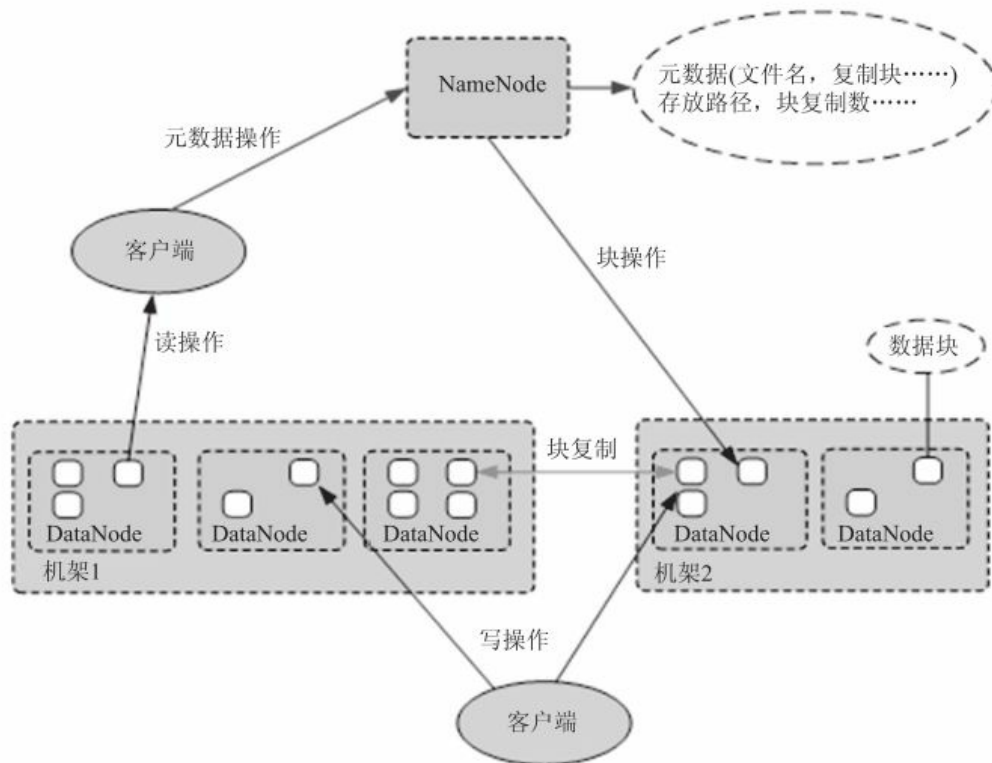


图10-5 HDFS工作原理图

在HDFS内部，一个文件可分为一个或多个“数据块”，这些“数据块”存储在DataNode集合里。客户端连接到NameNode，执行文件系统的“命名空间”操作，例如打开、关闭、重命名文件和目录，同时决定“数据块”到具体DataNode节点的映射。DataNode在NameNode的指挥下进行“数据块”的创建、删除和复制。

因为NameNode负责保管和管理所有的HDFS元数据，所以用户数据的读写就不需要通过NameNode，而是直接在DataNode上进行。

10.2.4 MapReduce

MapReduce是一种简化并行计算的编程模型，名字源于该模型中的两项核心操作：Map和Reduce。Map将一个任务分解成为多个任务，Reduce将分解后多任务处理的结果汇总起来，得出最终的分析结果。MapReduce模型主要由ResourceManager、ApplicationMaster和NodeManager组成，如图10-6所示。

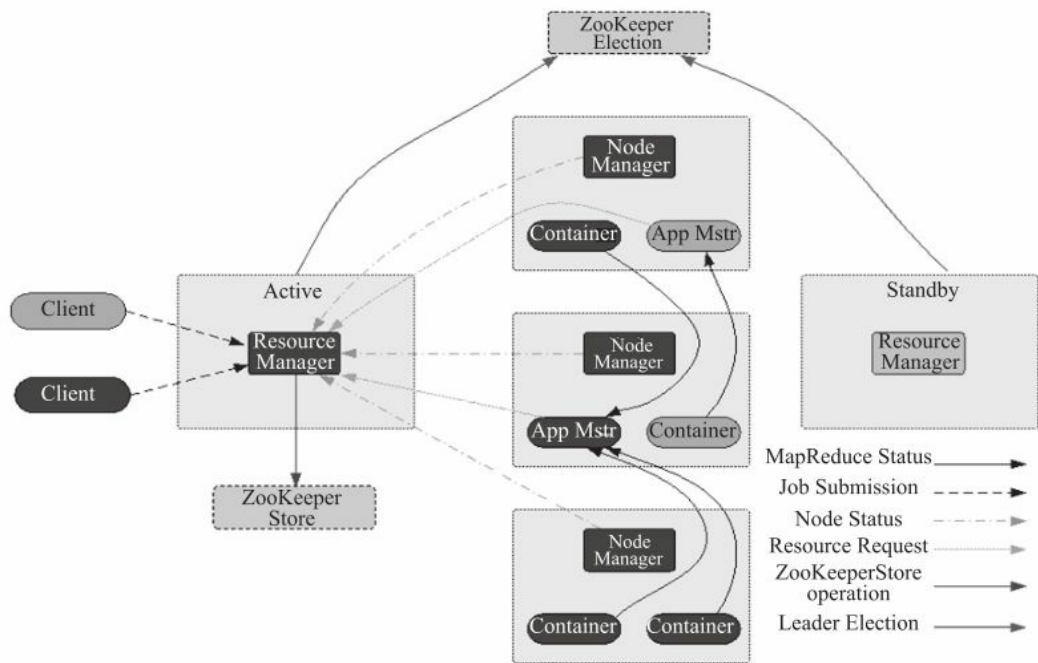


图10-6 MapReduce框架（基于Yarn）架构

MapReduce各部分的功能说明如表10-2所示。

表10-2 结构图说明

名称	描述
Client	MapReduce的客户端，可以通过客户端向ResourceManager发起MapReduce操作
ResourceManager	MapReduce的资源管理器，基于应用程序对资源的需求进行调度；由于每一个应用程序需要不同类型的资源，因此需要不同的容器；同时，资源管理

器提供一个调度策略的插件，它负责将集群资源分配给多个队列和应用程序；调度插件可以基于现有的能力调度和公平调度模型进行调度

NodeManager 负责执行应用程序的容器，同时监控应用程序的资源使用情况（CPU、内存、硬盘、网络），并向ResourceManager汇报

ApplicationMaster 负责相应的调度和协调，结合从ResourceManager获得的资源和NodeManager协同工作来运行和监控任务

Container 作为资源隔离，当前仅提供java虚拟机CPU、内存的隔离

新的Hadoop MapReduce框架命名为MapReduceV2或者Yarn。Hadoop MapReduce新框架主要分为ResourceManager、ApplicationMaster与NodeManager三个部分。

➤ ResourceManager核心服务，负责调度、启动每一个Job所属的ApplicationMaster、同时监控ApplicationMaster的运行情况。ResourceManager负责作业与资源的调度，并接收JobSubmitter提交的作业，按照作业的上下文（Context）信息，以及从NodeManager收集来的状态信息，启动调度过程，分配一个Container作为ApplicationMaster。

➤ NodeManager功能比较专一，就是负责Container状态的维护，并向ResourceManager保持心跳。

➤ ApplicationMaster负责一个Job生命周期内的所有工作，类似老框架中的JobTracker。但注意每一个Job（不是每一种）都有一个ApplicationMaster，它可以运行在ResourceManager以外的机器上。

10.2.5 ZooKeeper

ZooKeeper是一个分布式、高可用性的协调服务。产品化Hadoop系统中主要提供两个功能：

➤ 帮助系统避免单点故障，建立可靠的应用程序；

- 提供分布式协作服务和维护配置信息。

ZooKeeper集群中的节点分为三种角色，即Leader、Follower和Observer，其结构和相互关系如图10-7所示。

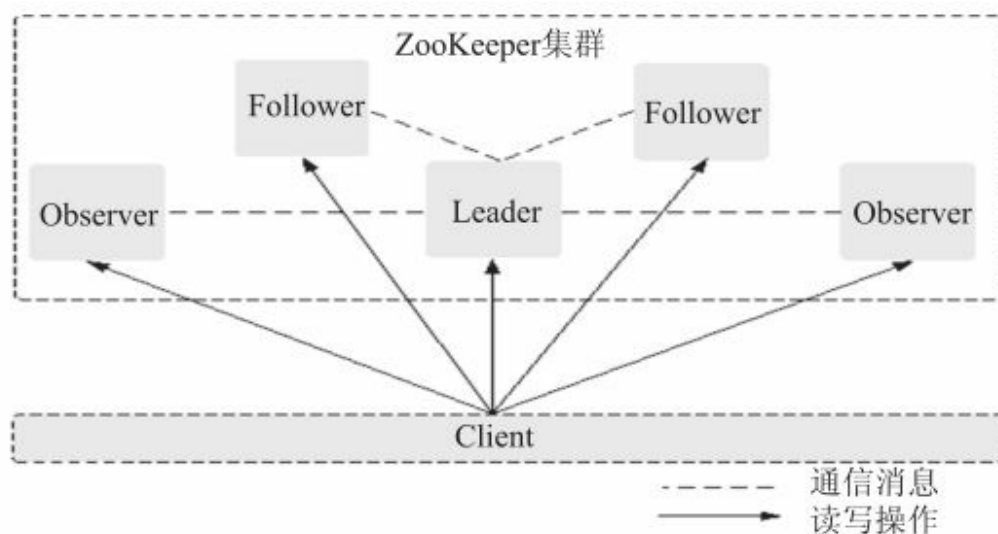


图10-7 ZooKeeper结构图

ZooKeeper各部分的功能说明如表10-3所示。

表10-3 ZooKeeper各部分的功能说明

名称	描述
Leader	在Zookeeper集群中只有一个节点作为集群的领导者，由各Follower通过Paxos算法选举产生，主要负责接受和协调所有写请求，并把写入的信息同步到Follower和Observer
Follower	Follower的功能有两个： <ul style="list-style-type: none">➤ 每个Follower都作为Leader的储备，当Leader故障时重新选举Leader，避免单点故障；➤ 配合Leader一起进行写请求处理
Observer	Observer不参与选举，负责接受写请求、处理读请求，避

免系统处理能力浪费

Client Zookeeper集群的客户端，对Zookeeper集群进行读写操作。例如HBase可以作为Zookeeper集群的客户端，利用Zookeeper集群的仲裁功能，控制其HMaster的“Active”和“Standby”状态

ZooKeeper写请求原理：

- Follower或Observer接受到写请求后，转发给Leader；
- Leader协调各Follower，通过投票机制决定是否接受该写请求；
- 投票结果为接受，则由Leader处理写请求；
- 数据写入完成后，Leader向Follower和Observer同步，保证所有节点的数据一致性。

ZooKeeper只读请求原理：客户端直接向Leader、Follower或Observer读取数据。

10.2.6 HBase

HBase是一种构建在HDFS（Hadoop Distributed File System）之上的分布式、面向列的存储系统，它具有高可靠、高性能、面向列和可伸缩的特性。HBase适合于存储大表数据（表的规模可以达到数十亿行以及数百万列），并且对大表数据的读、写访问可以达到实时级别。

HBase集群由主备HMaster进程和多个RegionServer进程组成，如图10-8所示。

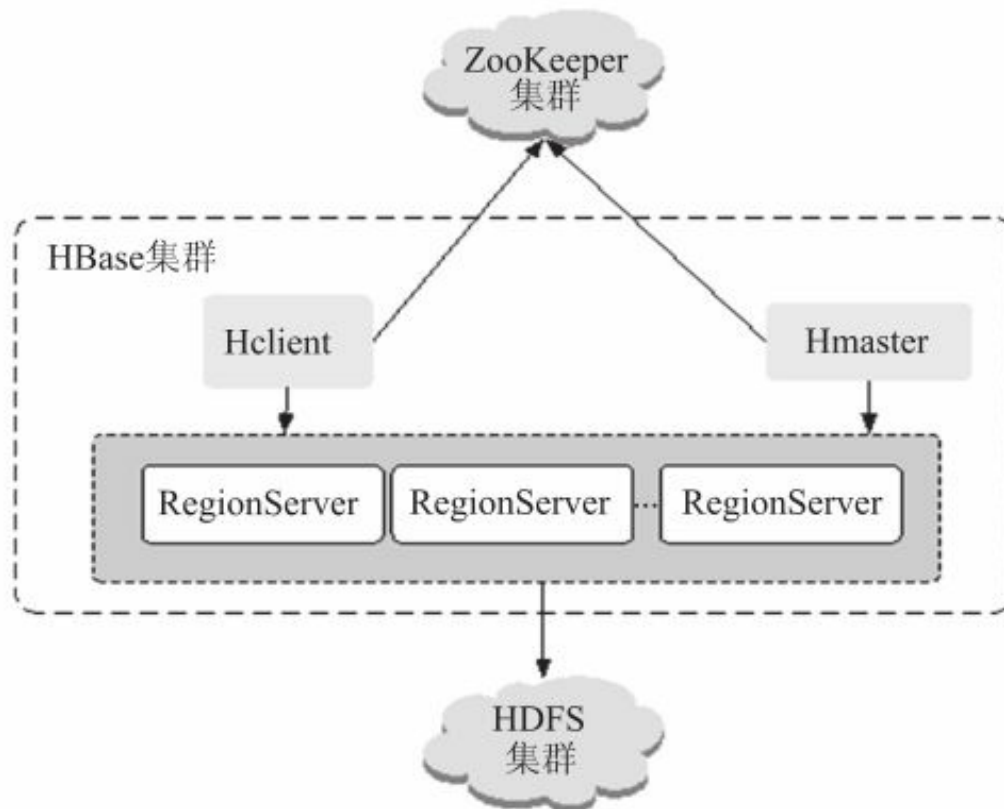


图10-8 HBase结构

HBase各部分的功能说明如表10-4所示。

表10-4 HBase功能说明

名称	描述
主用HMaster	主用HMaster负责管理RegionServer节点，同时维护集群的网络拓扑以及集群的负载均衡
备用HMaster	当主用HMaster故障时，备用HMaster将取代主用HMaster对外提供服务；故障恢复后，原主用HMaster降为备用
RegionServer	RegionServer负责提供表数据读写等服务，是HBase的数据处理和计算单元；RegionServer一般与HDFS集群的DataNode合设，实现数据的存储功能
ZooKeeper集	ZooKeeper为HBase集群中各进程提供分布式协作服

群 务；各RegionServer将自己的信息注册到Zookeeper中，HMaster据此感知各个RegionServer的健康状态

HDFS集群 HDFS为HBase提供高可靠的文件存储服务，HBase的数据全部存储在HDFS中

HBase以表的形式存储数据，数据模型如图10-9所示。表中的数据划分为多个Region，并由HMaster分配给对应的RegionServer进行管理，每个Region包含了表中一段Row Key区间范围内的数据，HBase的一张数据表开始只包含一个Region，随着表中数据的增多，当一个Region的大小达到容量上限后会分裂成两个Region。

Row Key	Timestamp	Column Family 1		Column Family N		
		URL	Content	Column 1	Column 2	
Row1	t2	www.huawei.com	"<html>"			Region
	t1	www.huawei.com	"<html>"			
...	
RowM						
RowM+1	t1	Region
RowM+2	t3	
	t2	
	t1	
...	
RowN	t1	Region
...	

图10-9 HBase数据模型

HBase数据模型中各列的说明如表10-5所示。

表10-5 数据模型列说明表

名称	描述
Row Key	表的主键，数据存储时表中的记录按照Row Key的字符序进行排序
Timestamp	插入数据时对应的时间戳，HBase支持相同Row Key的多

版本数据存储

Column Family 列族，HBase中表在水平方向上由一个或者多个Column Family组成，一个Column Family由任意多个Column组成

Column 列，与传统的数据库类似，HBase的表中也有列的概念，列用于表示相同类型的数据

RegionServer数据存储：RegionServer主要负责管理由HMaster分配的Region，RegionServer的数据存储结构如图10-10所示。

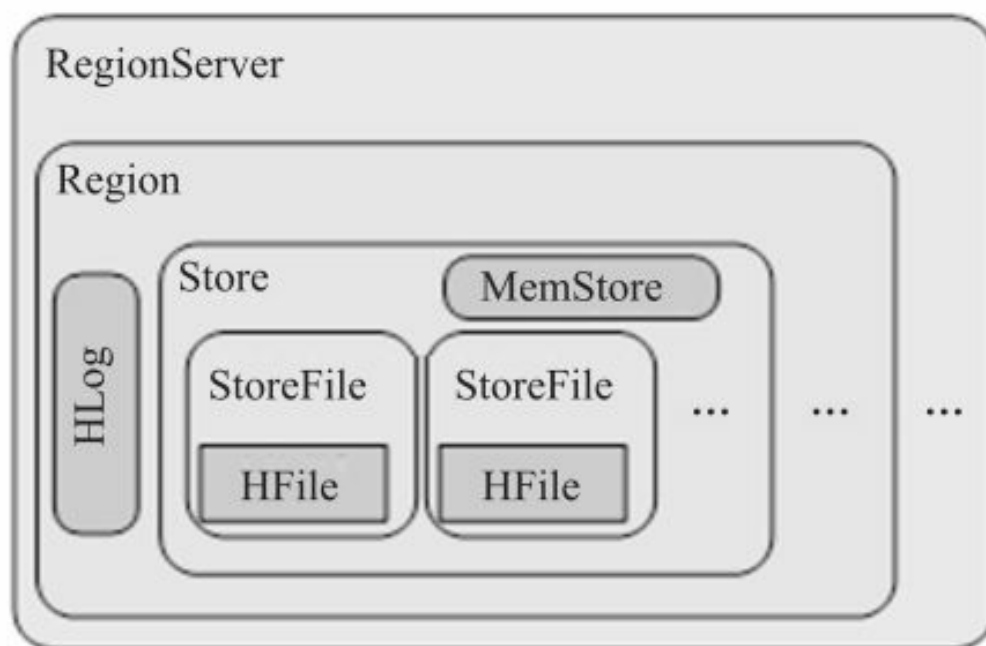


图10-10 RegionServer的数据存储结构

Region的各部分的说明如表10-6所示。

表10-6 Region结构说明

名称	描述
Store	一个Region由一个或多个Store组成，每个Store对应图10-10中的一个Column Family
MemStore	一个Store包含一个MemStore，MemStore缓存客户端向

Region插入的数据，当RegionServer中的MemStore大小达到配置的容量上限时，RegionServer会将MemStore中的数据“flush”到HDFS中

StoreFile MemStore的数据“flush”到HDFS后成为StoreFile，随着数据的插入，一个Store会产生多个StoreFile，当StoreFile的个数达到配置的最大值时，RegionServer会将多个StoreFile合并为一个大的StoreFile

HFile HFile定义了StoreFile在文件系统中的存储格式，它是当前HBase系统中StoreFile的具体实现

HLog HLog日志保证了当RegionServer发生故障时用户写入的数据不丢失，RegionServer的多个Region共享一个相同的HLog

元数据表是HBase中一种特殊的表，用来帮助Client定位到具体的Region，包括“.META.”表和“-ROOT-”表。

➤ “.META.”表：记录用户表的Region信息，例如Region位置、起始Row Key及结束Row Key等信息。

➤ “-ROOT-”表：记录“.META.”表的Region信息。“-ROOT-”表不会分裂，因此只有一个Region包含“-ROOT-”表。

元数据表和用户表的映射关系如图10-11所示。

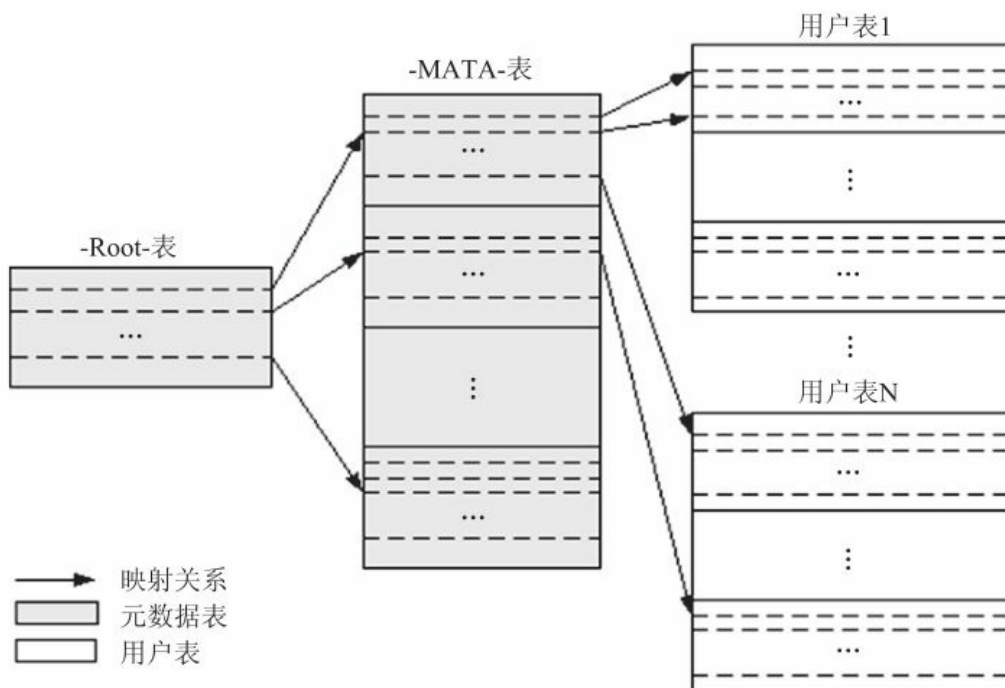


图10-11 元数据表和用户表的映射关系

数据操作流程

HBase数据操作流程如图10-12所示。

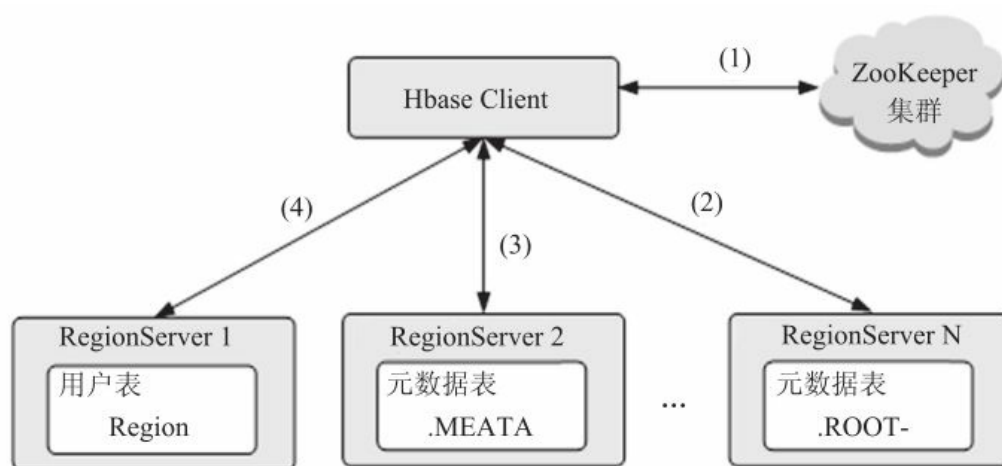


图10-12 HBase数据操作流程

(1) 当Hadoop对HBase进行增、删、改、查数据操作时，HBase Client首先连接ZooKeeper获得“-ROOT-”表所在的RegionServer的信息。

(2) HBase Client 连接到“-ROOT-”表的Region所在的RegionServer，并获得“.META.”表的Region的信息。

(3) HBase Client 连接到包含对应的“.META.”表的Region所在的RegionServer，并获得相应的用户表的Region所在的RegionServer位置信息。

(4) HBase Client 连接到对应的用户表Region所在的RegionServer，并将数据操作命令发送给该RegionServer，RegionServer接收并执行该命令，从而完成本次数据操作。

为了提升数据操作的效率，HBase Client 会在内存中缓存“-ROOT-”、“.META.”和用户表Region的信息，当应用程序发起下一次数据操作时，HBase Client 会首先从内存中获取这些信息；当内存中缓存的数据信息与系统中的实际信息不一致时，HBase Client 会重复上述操作。

10.2.7 Hive

Hive是一个基于Hadoop的开源的数据仓库平台，通过Hive，可以方便地进行数据提取转化加载（ETL）的工作，提供类似SQL的HQL语言，操作结构化数据存储服务和基本的数据分析服务。HQL能够将用户编写的SQL转化为相应的MapReduce程序。当然，用户也可以自定义Mapper和Reducer来完成复杂的分析工作。基于MapReduce的Hive具有良好的扩展性和容错性。不过由于MapReduce缺乏结构化数据分析中有价值的特性，以及Hive缺乏对执行计划的充分优化，导致Hive在很多场景下比并行数据仓库慢（在几十台机器的小规模下可能相差更大）。

Hive主要特点如下：

- 海量结构化数据分析汇总；
- 将复杂的MapReduce编写任务简化为SQL语句；
- 灵活的数据存储格式，支持JSON、CSV、TEXTFILE、RCFILE、SEQUENCEFILE这几种存储格式。

为保证Hive服务的高可用性、用户数据的安全及访问服务的可控制性，

在开源社区的Hive-0.9.0版本上新增双机特性和安全特性。

开源社区的Hive特性，请参见

<https://cwiki.apache.org/confluence/display/Hive/DesignDocs>。

Hive双机特性采用基于双机设备，即基于主备切换方式的服务器设备来保证Hive服务的高可用性。在同一时间内只有一个Hive Server运行，当主Hive Server出现故障无法继续提供服务时，备Hive Server会被激活，保证业务在短时间内完全恢复正常使用。

通过Zookeeper的Master Election机制保证总有一个Hive Server正常运行并提供服务。主备两个HiveServer以Ephemeral方式分别注册到Zookeeper中，使得其中一个Hive Server出现故障时，能够及时选举并恢复服务。同时，Zookeeper中存储主Hive Server的IP地址。Hive客户端可通过Zookeeper获取主Hive Server的IP地址来连接Hive Server以获得服务。

在双机环境下，需要知道Zookeeper的IP地址和端口才能获取主Hive Server的地址；在安全环境下，除了需要Zookeeper的IP地址和端口信息外，还需要连接Zookeeper的相关认证信息。

Hive用户认证采用的是基于Kerberos的认证。Kerberos是一种适用于在公共网络上进行分布计算的工业标准的安全认证系统。用户通过Hive客户端和Hive Server建立连接时，需要进行双向认证。当用户是Kerberos KDC的合法用户时，其才能够通过认证，访问Hive的服务。认证方法为用户使用用户名（Principal）及Keytab文件登录KDC，登录成功，则表示认证通过。

Hive使用user、group、role对权限进行管理。在Hive中，进行数据定义、加载和查询都需要相应的操作权限。

Hive作为强大的数据仓库和数据分析平台至少需要具备以下几点特性：

- 灵活的存储引擎；
- 高效的执行引擎；
- 性能良好的索引机制；

- 良好的可扩展性；
- 强大的容错机制；
- 多样化的可视化。

先看看Hive是否完全具备了以上几点，以及传统的并行数据仓库对比优劣如何。

存储引擎

Hive没有自己专门的数据存储格式，也没有为数据建立索引，用户可以非常自由地组织Hive中的表，只要在创建表时告诉Hive数据中的列分隔符和行分隔符，Hive就可以解析数据。Hive的元数据存储于RDBMS中，所有数据都基于HDFS存储。Hive包含Table、External Table、Partition和Bucket等数据模型。

并行数据仓库需要先把数据装载到数据库中，按特定的格式存储，然后才能执行查询。每天需要花费几个小时将数据导入并行数据库中，而且随着数据量的增长和新的数据源加入，导入时间会越来越长。导入时大量的写I/O与用户查询的读I/O产生竞争，会导致查询的性能很差。

Hive执行查询前无需导入数据，直接执行计划。Hive支持默认的多种文件格式，同时可以通过实现MapReduce的InputFormat或OutputFormat类，由用户定制格式。因为公司的数据种类很多，存储于不同的数据源系统，如MySQL、HDFS或者Hypertable等，很多时候Hive的分析过程会用到各种数据源的数据。当然使用多个存储数据源，除了功能上要能够支持导入/导出之外，如何根据各种存储源的能力和流获得最优执行计划也是一件麻烦的事情。

执行引擎

MR对于Map和Reduce job间的数据传输处理方式具有潜在的性能问题。假设有N个Map instance，每个产生M个输出文件，每个输出文件指向不同的Reduce instance。这些文件会被写到执行Map instance节点的本地磁盘。如果N是1000，M是500，那么Map阶段将会产生500 000个本地文件。当Reduce阶段开始时，500个Reduce instance中每个都需要读取1000个输入文件，同时必须使用一个文件传输协议从每个Map instance所运

行的节点处拉取输入文件。假设同时有100个Reduce instance并行执行，不可避免地将会有两个或者更多的Reduce instance同时在同一个节点上试图读取文件，这就会产生大量的磁盘seek操作，从而降低磁盘传输效率。这也是为什么并行数据库系统没有将它们的中间数据保存为文件，并且采用了一种推模式而不是拉模式进行数据传输的原因。

并行数据仓库使用优化器进行性能优化。在生成执行计划时，利用元数据信息估算执行流上各个算子要处理的数据量 and 处理开销，进而选取最优的执行计划。并行数据仓库实现了各种执行算子（Sort、GroupBy、Union和Filter等）。优化器可以灵活地选择这多个算子以实现不同类别的性能优化。此外，并行数据仓库还拥有完备的索引机制，包括磁盘布局、缓存管理和I/O管理等多个层面的优化，这些都对查询性能至关重要，而这恰恰是Hive的不足之处。

Hive的编译器负责编译源代码并生成最终的执行计划，包括语法分析、语义分析、目标代码生成，所做的优化并不多。Hive基于MapReduce，Hive的Sort和GroupBy都依赖MapReduce。而MapReduce相当于固化了执行算子，Map的MergeSort必须执行，GroupBy算子也只有一种模式，Reduce的Merge-Sort也必须可选。另外Hive对Join算子的支持也较少。另外，内存复制和数据预处理也会影响Hive的执行效率。当然，数据预处理可能会影响数据的导入效率，这需要根据应用特点进行权衡。

索引机制

所有的现代DBMS都使用hash或者B树索引来加速数据访问。如果某人要查找一个记录子集（比如工资大于100 000美元的雇员），那么使用合适的索引可以明显缩小查询的范围。大部分数据库都允许单个表格具有多个索引。因此，查询优化器可以决定为用户查询使用哪个索引或者是简单地采用一个暴力的顺序搜索。

Hive在加载数据的过程中不会对数据进行任何处理，甚至不会对数据进行扫描，因此也没有对数据中的某些Key建立索引。Hive要访问数据中满足条件的特定值时，需要暴力扫描整个数据，因此访问延迟较高。由于MapReduce的引入，Hive可以并行访问数据，即使没有索引，对于大数据量的访问，Hive仍然可以体现出优势。由于数据的访问延迟较高，决定了Hive不适合在线数据查询。

扩展性

并行数据仓库可以很好地扩展到几十或上百个节点的集群，并且达到接近线性的加速比。然而，今天的大数据分析需要的可扩展性远远超过这个数量，经常需要达到数百甚至上千节点。目前，几乎没有哪个并行数据仓库运行在这么大规模的集群上，这涉及多个方面的原因。并行数据仓库假设底层集群节点完全同构；并行数据仓库认为节点故障是很少出现的；并行数据仓库设计和实现基于的数据量并未达到PB级或者EB级。

与并行数据仓库不同的是，Hive更加关注水平扩展性。简单来讲，水平扩展性指系统可以通过简单地增加资源来支持更大的数据量和负载。

Hive处理的数据量是PB级的，而且每小时每天都在增长，这就使得水平扩展性成为一个非常重要的指标。Hadoop系统的水平扩展性是非常好的，基于MapReduce框架，Hive能够很自然地利用这一点。

容错性

Hive具有较好的容错性。Hive的执行计划在MapReduce框架上以作业的方式执行，每个作业的中间结果文件写到本地磁盘，最终输出文件写到HDFS文件系统，利用HDFS的多副本机制来保证数据的可靠性，从而达到作业的容错性。如果在作业执行过程中某节点出现故障，那么Hive执行计划基本不会受到影响。因此，基于Hive实现的数据仓库可以部署在由普通机器构建的分布式集群之上。

当某个执行计划在并行数据仓库上运行时，若某节点发生故障，则必须重新执行该计划。所以，当集群中的单点故障（可能是磁盘故障等）发生率较高时，并行数据仓库的性能就会下降。在实际生产环境中，假设每个节点故障发生率是0.01%，那么1000个节点的集群中单点故障发生率则为10%。这个数字并非耸人听闻，处理海量数据的I/O密集型应用集群，平均每月的机器故障率达到1%~10%。当然这些机器可能是2万~3万元的普通机型。

可视化

Hive的可视化界面基本属于字符终端，用户的技术水平一般比较高。面向不同的应用和用户，提供个性化的可视化展现，是Hive改进的一个重要方向。个性化的可视化也可以理解为用户群体的分层，例如图形界面方式提供初级用户，简单语言方式提供中级用户，复杂程序方式提供高

级用户。

10.3 流处理技术

10.3.1 流处理的应用场景

对大数据技术不熟悉的人而言，流处理技术最容易让人联想到的是视频和音频这种流媒体实时处理，比如视频监控，一边拍着视频，一边对视频内容做实时的分析处理，比如在城市建设里，通过摄像头能动态查找犯罪嫌疑人。在反恐动作片里，利用大数据分析技术自动定位、查找出恐怖分子。很遗憾，目前大数据分析技术中，流处理技术还没有这么先进。之所以没有达到这么理想的效果，也不都是大数据分析处理的问题，除了大数据分析处理还不够强大外，配套设施也不是很先进，比如我们的摄像头还无法把在摄像范围内的人脸拍得那么清晰，特别是现在雾霾比较严重的情况下，摄像头的表现并没有比人眼出色多少。模糊处理技术也还不够智能，当前的数据分析最多能把活动目标进行锁定（而且还是在夜深人静、干扰很少的情况下），并判断其活动轨迹和活跃度（速度），至于这个活动目标是不是嫌疑人，还得求助警察通过进一步的工作来完成。

当前的流处理技术，更多地侧重于结构化、半结构化的文本类文件，如日志类文件的处理，其特征是：事件驱动、实时处理（数据先计算，不存储或后存储）、毫秒/秒级（分析响应速度要求）、实时监控、高级事件触发、立即事件浅加工。

流处理技术能解决哪些问题呢？下面举几个例子：

- 对于电信运营商：当前网络质量是什么状态？有什么异常点发生？
- 对于券商：当前是否可以买入或者卖出一支股票？
- 对于信用卡银行：当前是否发生了异常、欺诈交易？可以给用户提供什么推荐服务？
- 对于电商：客户正在购物，可以给其推荐什么商品？

➤ 对于社交媒体：用户当前的访问模型是什么？当前热点话题是什么？用户的类型、兴趣是什么？

➤ 对于城市道路交通系统：当前各道路流量、车速情况有什么异常，需要采取什么措施？

➤ 对于物流企业：当前货物库存、周转、运输状态是什么？有没有异常？

.....

10.3.2 流处理技术的关键概念

流处理技术利用时间“窗口”概念来让“流”中的数据有了“边界”。窗口中的数据等于“数据库中的一张静态表”（见图10-13）。

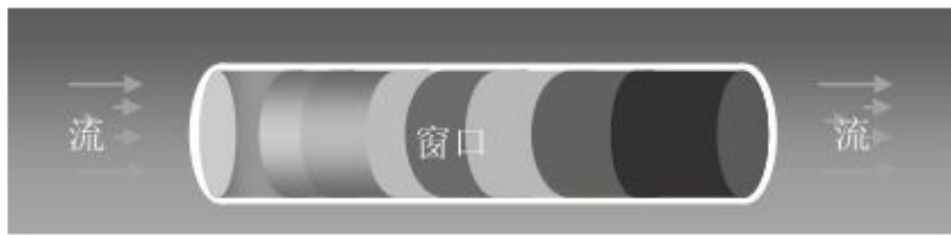
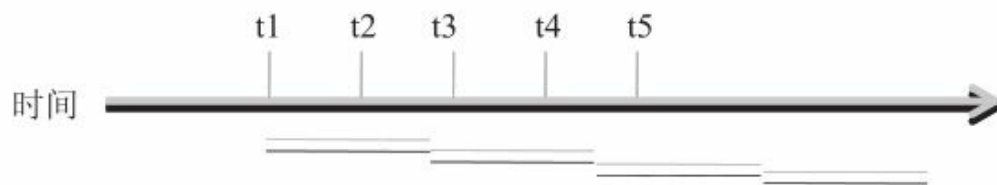


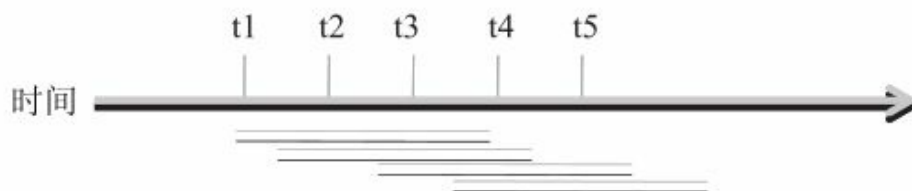
图10-13 窗口概念

“窗口”又可分为跳动窗口和滑动窗口，用于不同场景的分析处理（见图10-14、图10-15）。



跳动窗口(Jump window)

图10-14 跳动窗口



滑动窗口(Slide window)

图10-15 滑动窗口

例如：在电信业务中，其可以利用窗口的概念来进行数据流分析。电信信令监控的某业务是否是恶意呼叫，需寻找在连续1分钟内发起大于5次呼叫的手机用户（见图10-16），语句如下：

```
SELECT * FROM CallEvent.win:time (60 sec) GROUP BY strImsi
HAVING count (*) >5
```

在流处理中，这个查询语句是持续作用的，流入的事件满足条件即被触发持续查询，这一特性叫“持续查询”。

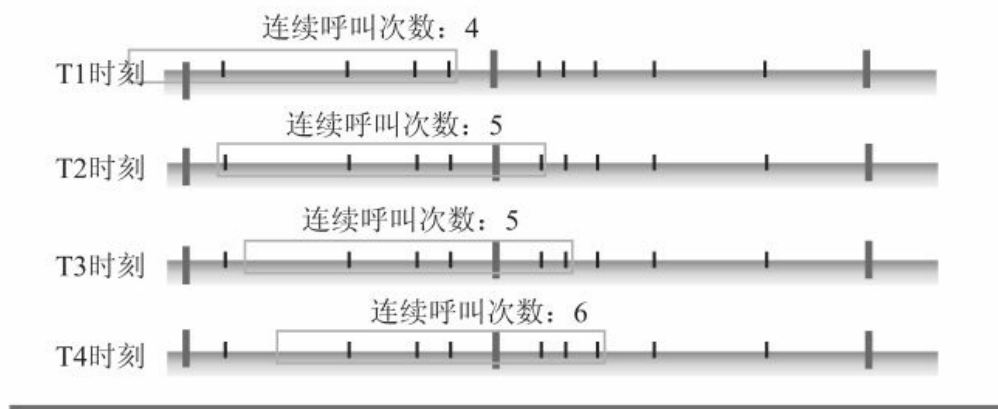


图10-16 恶意呼叫查询

通过查询处理，分析并发现数据反映出的关系和问题，叫“模式发现”，可以采用以下分析方式“（模式）发现”（见图10-17）：

- 跟随模式，A事件发生后总会发生B事件；
- 非事件，不发生某个事件；
- 周期性发生，A事件总是在某一时间段内发生。

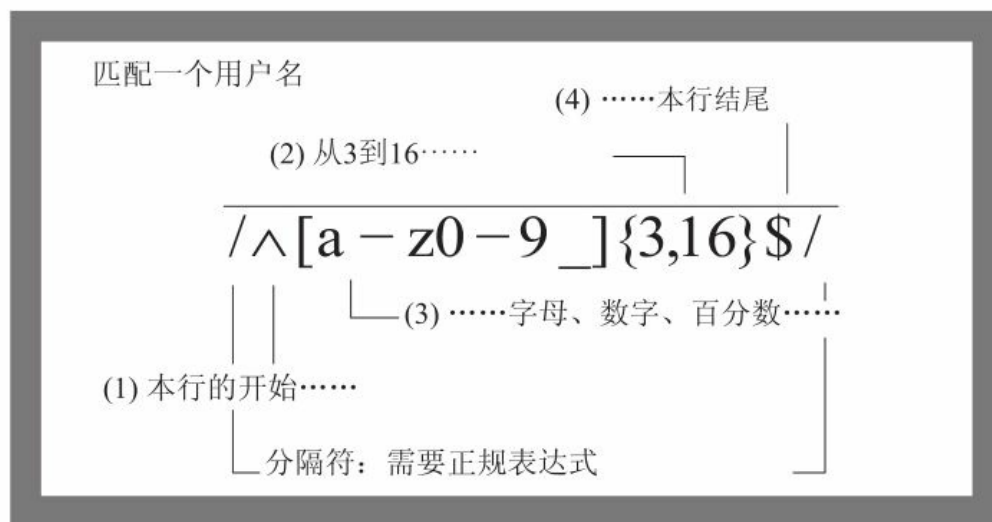


图10-17 文本搜索中的正则表达式

通过以上的文本搜索，获得以下分析结果，事件A发生后3秒内B或C跟随发生，在10秒时间内D不会发生，从而发现其中的问题（见图10-18）。

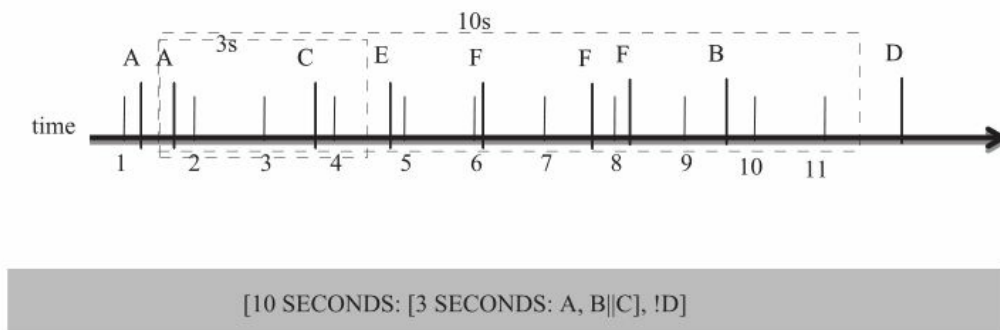


图10-18 异常发现

10.3.3 流处理技术的辨析

目前流处理技术主要有两种：CEP（Complex Event Processing）用于复杂事件处理；Stream Processing用于流处理。两种技术的对比如图10-19所示。

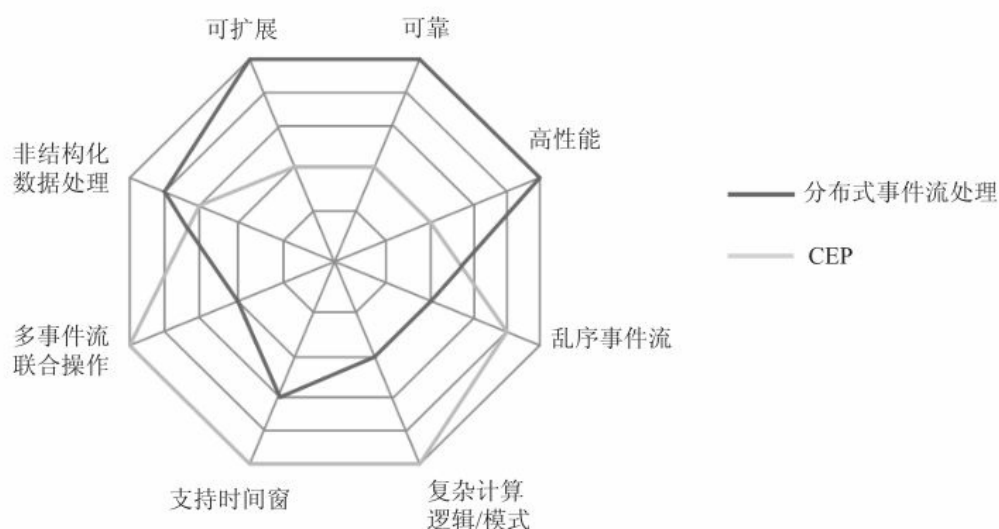


图10-19 流处理技术对比

CEP技术又分为基于查询和基于规则两种框架。基于查询（Query-Based）的CEP技术适用于高速数据流分析，基于规则的CEP技术则是“条件—规则—行动”的模式。

基于查询（Query-Based）的CEP技术框架如图10-20所示。

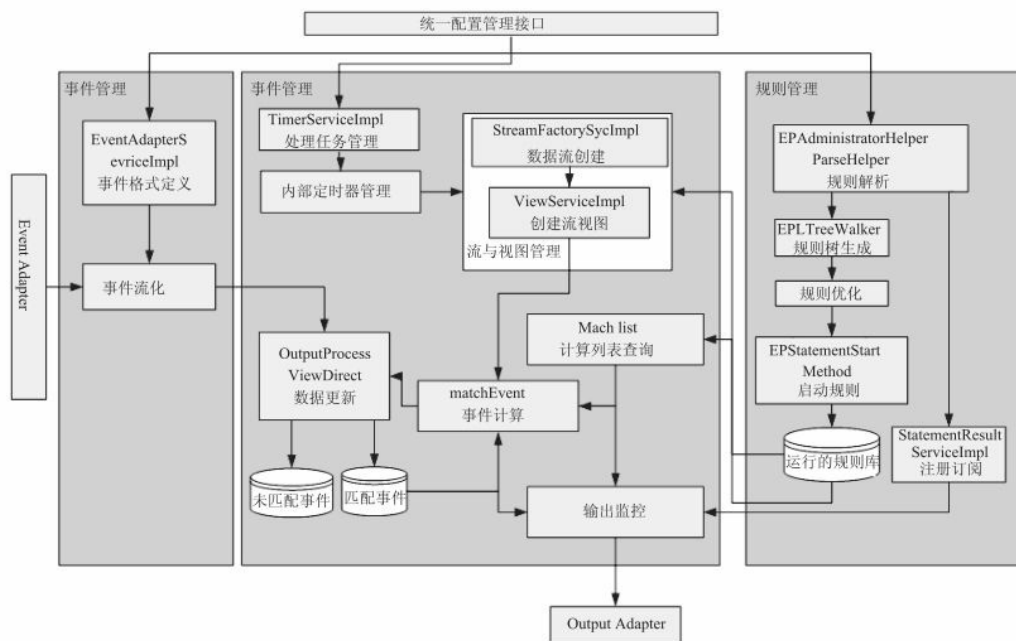


图10-20 基于查询的CEP技术框架

代码样例如图10-21所示。

```
select sum(price) from StockTickEvent(symbol='GE') win:length(5)
```

图10-21 代码样例

基于查询（Query-Based）的CEP技术基本特征：

- 支持基于类SQL查询语言；
- Antlr解析查询语句；
- 根据解析结果生成数据结构；
- 计算事件是否满足触发条件；
- 可以支持较高数据量。

基于查询（Query-Based）的CEP技术适用场景：

- 数据量非巨量，使用单节点可以满足需求；
- 较复杂类SQL业务，如KPI统计、定时输出结果、多流Join处理等。

基于规则（Rule-based）的CEP技术基本特征是：实时决策支持、数据吞吐量不大、基于“条件—规则—行动”、常用于BPM，以及可作为规则引擎、决策引擎。其技术框架如图10-22所示。

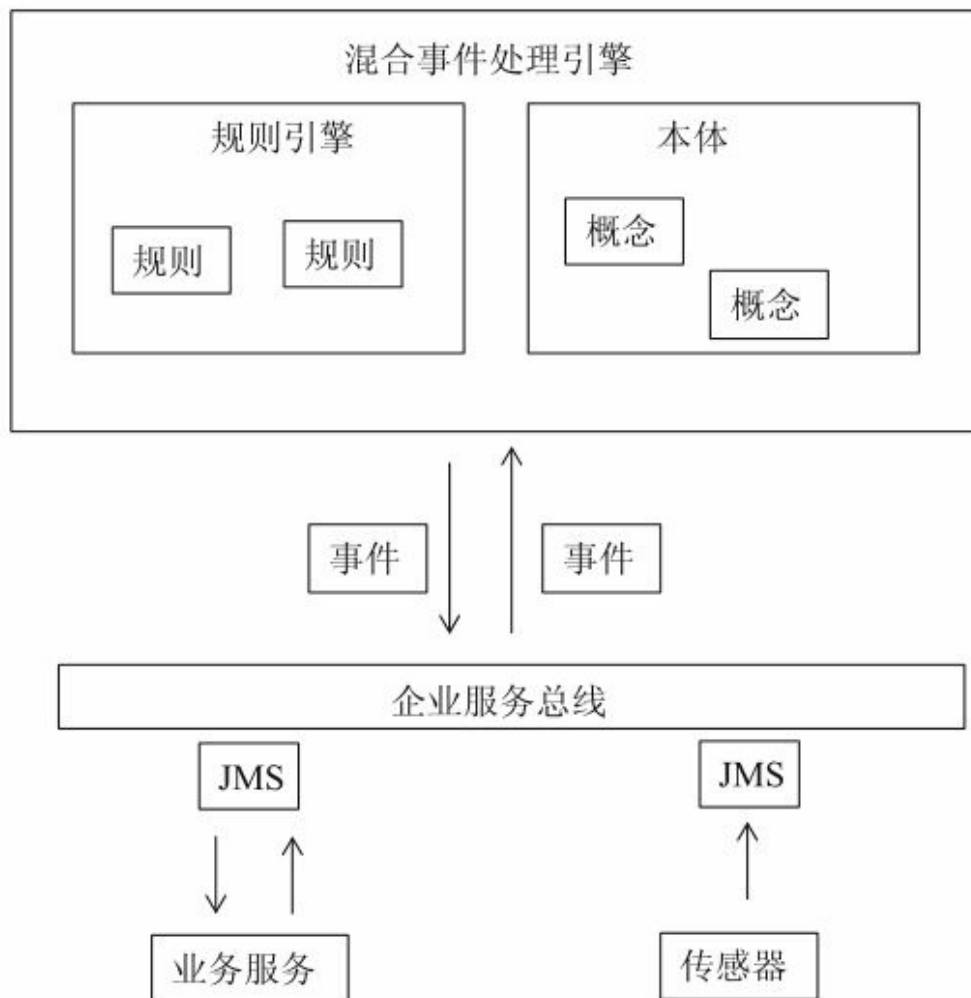


图10-22 基于规则的CEP技术框架

CEP技术发展有十几年的历史，近年借助大数据有升温趋势，特别是在金融证券、网络安全、物流、交通、物联网领域应用。CEP技术分类有Query-Based、Rule-Based、State-Based等，主流是Query-Based，也称为ESP（Event Stream Processing）。CEP在尝试标准化流、窗口、关系等

概念已写入SQL99规范，CQL（Continuous Query Language）将成为未来标准。但CEP未发展成SQL一样严谨、完备的理论体系，各家实现标准不一，开放性差。对于复杂业务，基于CEP进行开发很困难。而且CPE的Scale-out扩展能力也很受限制。

10.3.4 流处理技术的最新发展

最近兴起的流处理技术新动向，是基于Hadoop的MapReduce框架进行流处理技术改进，主要包括：HOP（Hadoop Online Prototype）、Hstreaming、基于Spark的Spark Streaming这几个分支。

Hop是Berkeley大学、Yahoo合作改进Hadoop的开源项目，Hop改造了Map和Reduce之间的Shuffer、Pipeline架构（Push模型），采用管道通信，支持实时类业务（如持续计算、在线聚合），并兼容Hadoop API（见图10-23）。

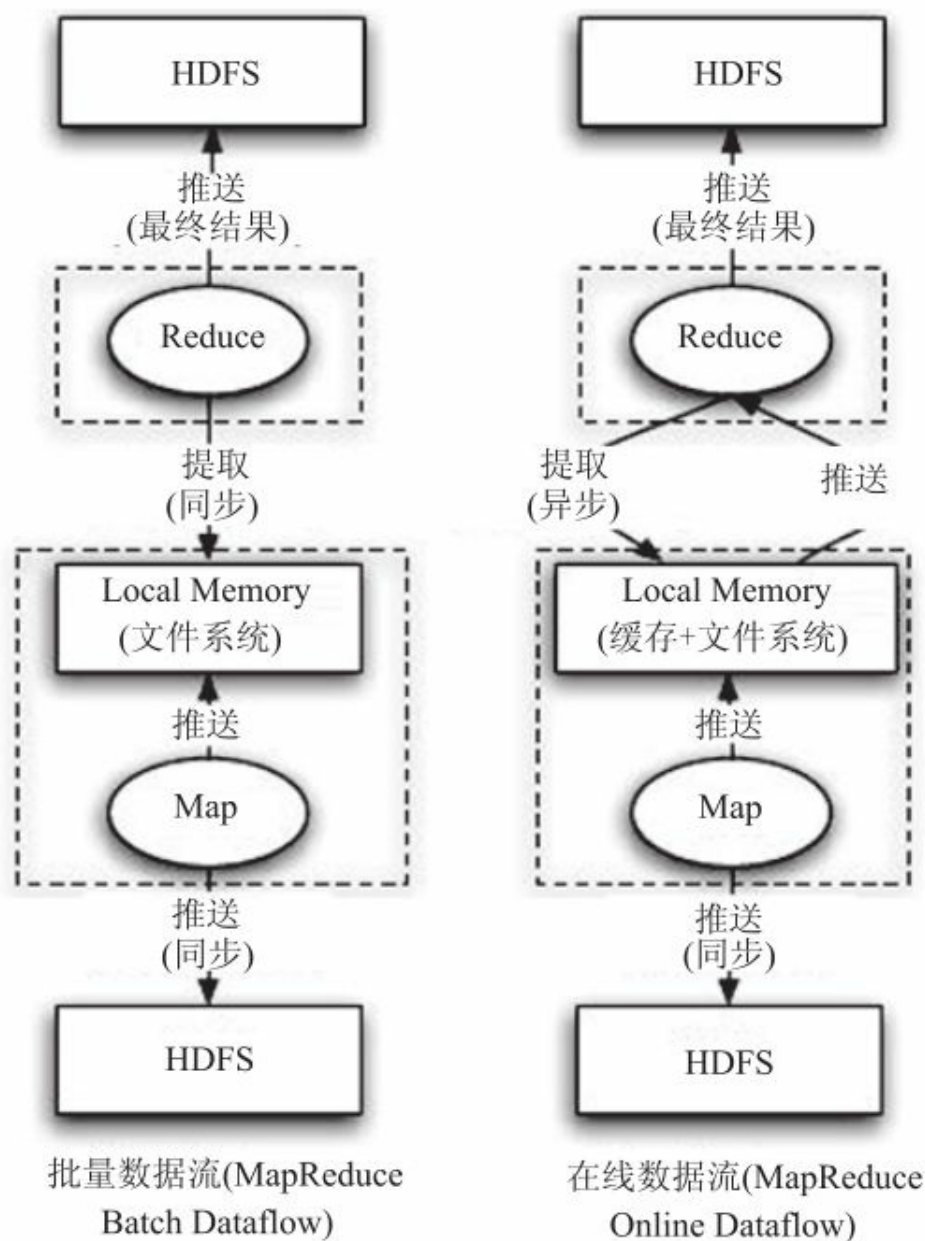


图10-23 基于Hadoop流处理模型

HStreaming基于Hadoop实现，采用HTTP/TCP通信，数据不落地，提供类似Pig的脚本语言提供流处理能力，包括流、窗口、聚合操作（见图10-24、图10-25）。

Spark Streaming是基于Spark实现的流处理，实现秒级延迟，支持流、窗口、各种聚合操作。Spark是一种内存集群计算框架（In-memory Cluster Computing Framework），用于应用系统复用数据。Spark适合迭代计算

（ML,DM），是一款“基于内存”的Hadoop技术（见图10-26、图10-27）。

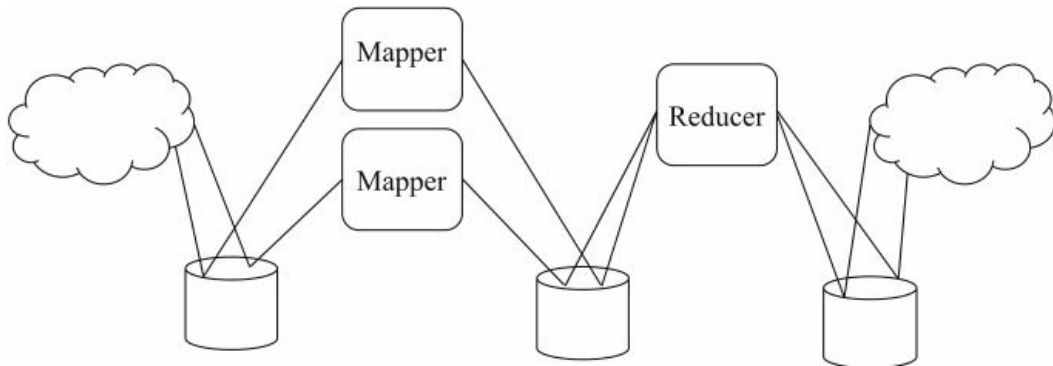


图10-24 MapReduce框架

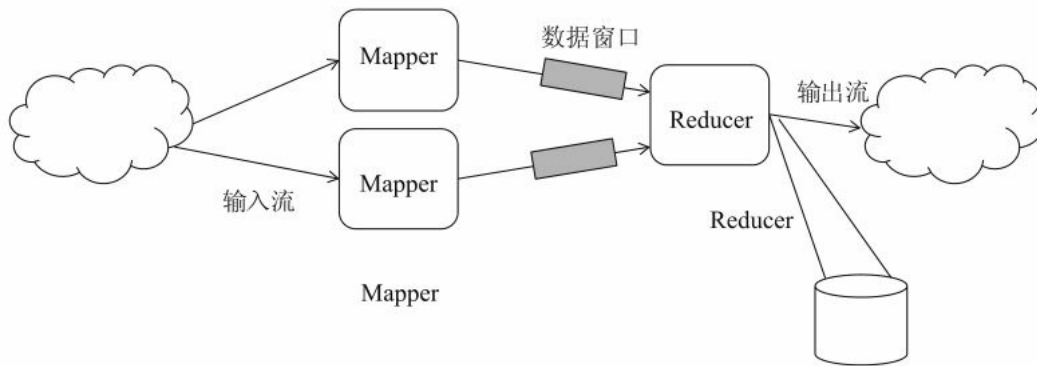


图10-25 HStreaming框架

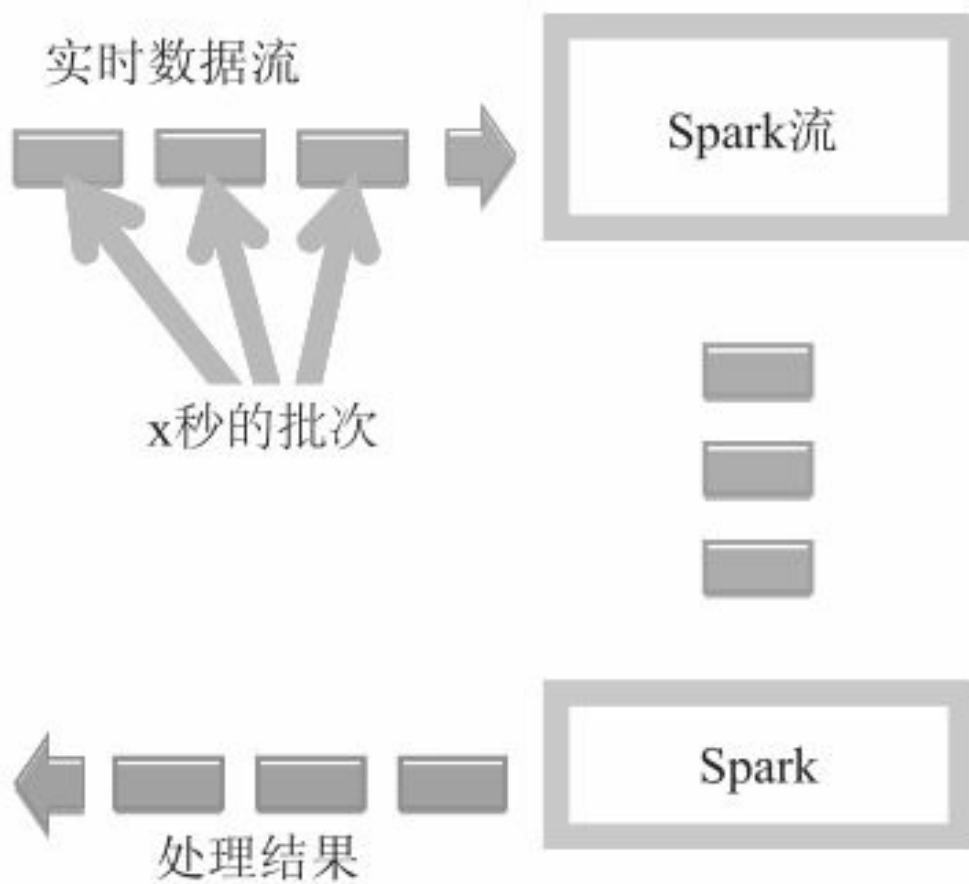


图10-26 Spark Streaming框架

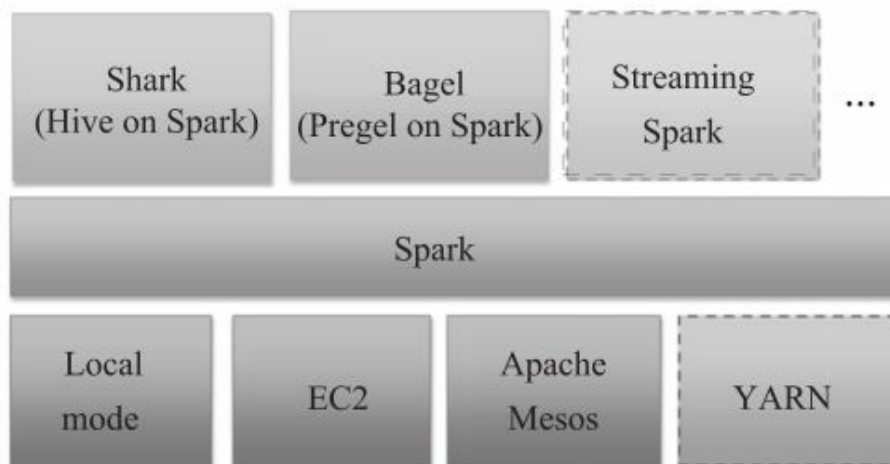


图10-27 Spark框架

基于Hadoop实现的各种流处理技术都是Pipeline方式的分布式处理框架，在Hadoop经典的MR模型下，修改任务调度、数据存储发送方式，

将MR的大批次改小，小到分钟级、秒级，这些技术部分兼容Hadoop API，作为Hadoop的补充，同时借助Hadoop生态系统，取得了一席之地。

10.3.5 分布式事件的流处理技术

在分布式事件流处理技术领域，主要是S4（Simple Scalable Streaming System）和Storm两种流处理技术。S4是Yahoo发起主导的Apache孵化项目，具有基于通用硬件、分布式、容错、可嵌入软件平台的特点，开发者可以很容易地开发应用来处理分析连续的流数据内容（见图10-28、图10-29）。

S4框架模型的特性包括事件驱动（Event-Driven）、Actor模型、DAG图的事件拓扑、用户自定义拓扑和处理逻辑、全分布式架构、对等结构，每个PN上有完整PE拓扑、支持采用Yarn部署、支持Fail-over、Checkpoint等。

Storm是由Twitter主导开发的分布式实时计算系统，目前拥护者众多，版本演进迅速（见图10-30）。

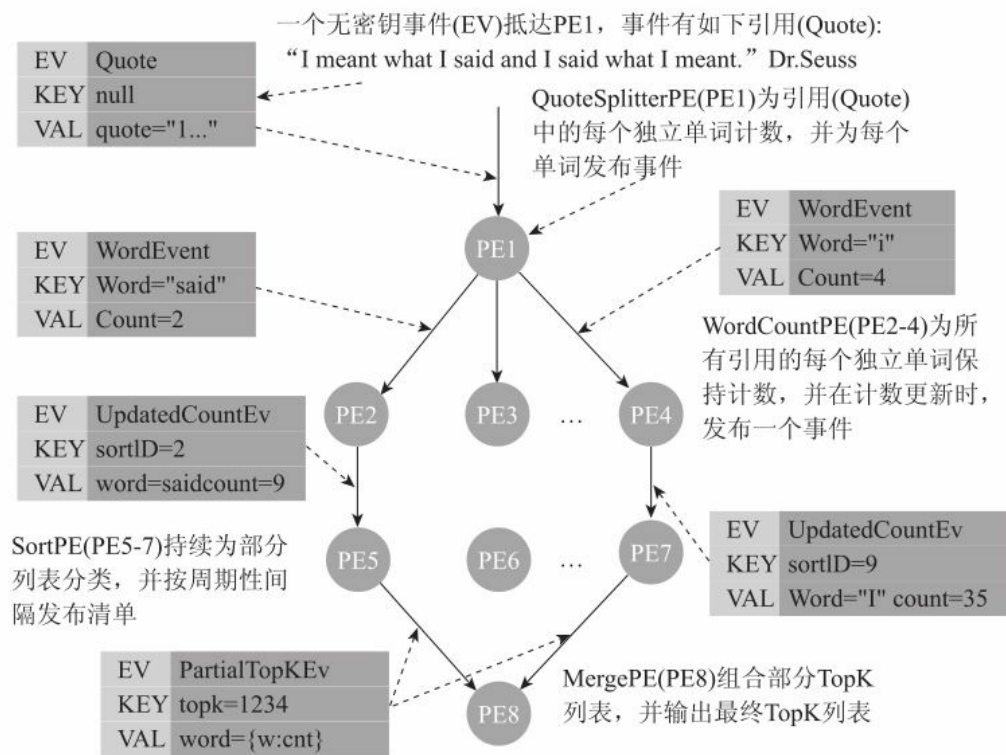


图10-28 S4流处理流程

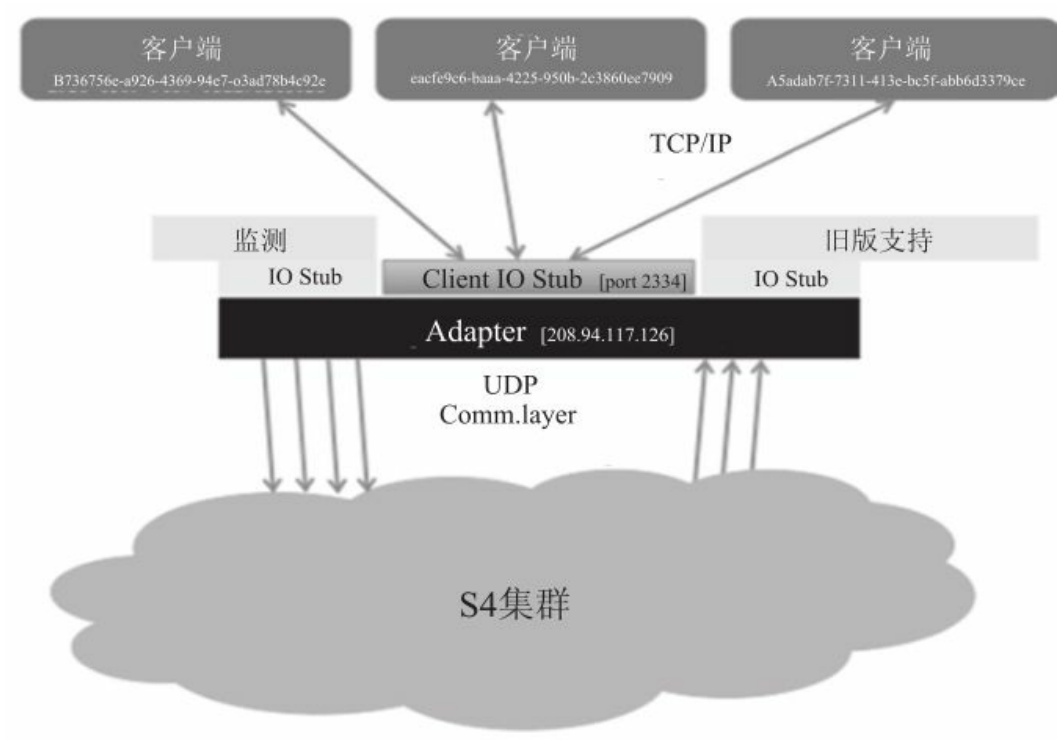


图10-29 S4流处理框架

Stream

以Tuple为基本单位组成的
一条有向无界的数据流



Topology

由计算节点和流动的
Tuple组成的拓扑结构

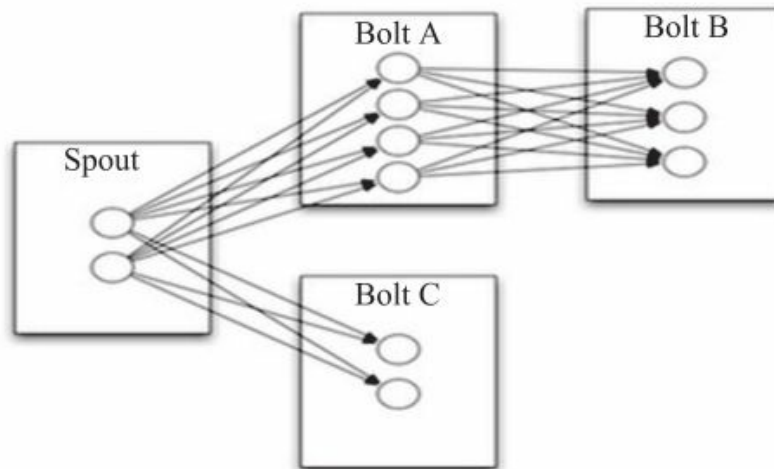
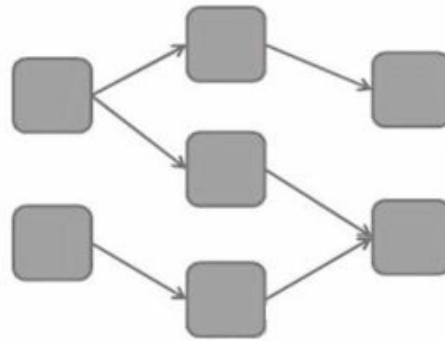


图10-30 Storm流处理框架

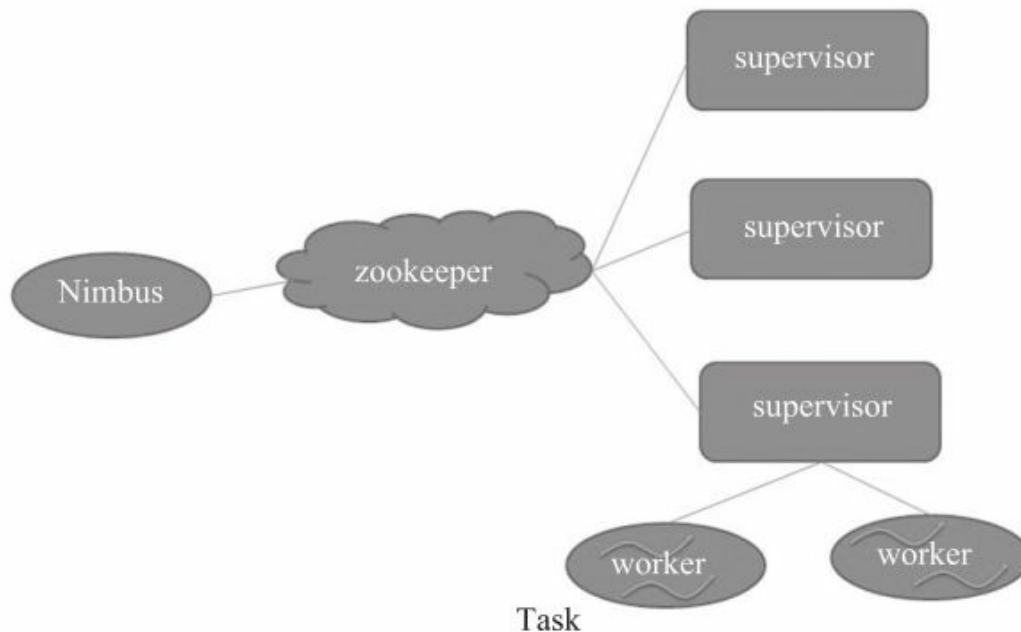


图10-30 Storm流处理框架（续）

Storm具有分布式处理框架，有中心管理节点，采用多种形式的事件路由分发机制，由事件驱动（Event-Driven），具有DAG图形式Topology，并支持事件处理的可靠性。

分布式事件流处理技术是真正的事件驱动（Event-driven）实时系统，分布式流处理引擎相当于Hadoop的MapReduce，可基于它实现上层的查询引擎、规则引擎、流挖掘等衍生的生态系统，有着巨大的想象空间。此外，各类相似类型的项目不断涌现，如Facebook Puma、Twitter Rainbird、LinkedIn Kafka、百度的Dstream，阿里巴巴的Galaxy、iProcess、SuperMario等。

10.4 大数据在金融领域的探索与实践

大数据在信息时代占据很高的地位，分析大数据对于众多行业都具有很重要的战略意义。大数据技术的主要任务是从内部和外部数据源中快速地发现商业机会并挖掘其价值，还对这些数据进行高效快捷的评估，最终提供决策支撑。大数据有利于从各类数据中快速获得信息，物联网的快速发展也为大数据提供了广泛的数据来源。智能手机的普及使数据量

呈指数级增长。廉价的存储和高速的带宽也为大数据的诞生提供了必备条件。云计算为大数据的诞生提供了物质基础。目前，数据量的增长在大数据应用中处于主导性的地位，从TB级升至PB级，并且仍在持续爆炸式增长，其增长速度远超摩尔定律增长速度。大数据的分析需求由抽样分析转向全量分析，成为企业发展必不可少的支撑点。由于数据量不断迅速增加，数据规模也随之增大，导致成本上升。从成本角度考虑，大数据的硬件平台由专用硬件服务器转向标准开放硬件构成的大规模机群平台。大数据遍布在许多领域，物联网、云计算、移动互联网、车联网、手机以及各式传感器，无一不是数据来源或者承载的方式。全球对大数据技术和服务的投资在不断增长，依靠大数据可以提供足够有利的资源。

大数据之于金融，其根本驱动是降低资本融通的成本、提高资金服务的效率（见图10-31）。

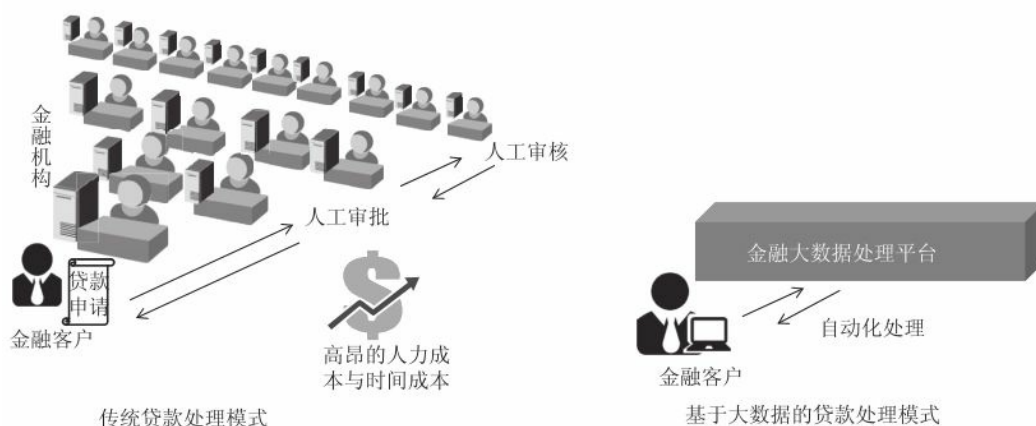


图10-31 大数据在金融行业

10.4.1 银行业现状和大数据的潜在机会

由于金融服务的业务转型的影响，银行业服务及管理模式都将发生根本性的改变。统计显示，以ATM、网上银行、手机银行为代表的电子银行在我国当前已经逐渐成为重要交易渠道，对传统银行渠道的替代率超过了60%。特别是互联网金融可能会对银行的观念和经营模式加以颠覆，银行业应如何主动变革、变挑战为机遇是一个值得探讨和深刻思考的问题。银行是经营信用的企业，数据的力量尤为关键和重要。在“大数据”时代，以互联网为代表的现代信息科技，特别是门户网站、社区论坛、微博、微信等新型传播方式的蓬勃发展，移动支付、搜索引擎和

云计算的广泛应用，构建起了全新的数字化客户信息体系，并将改变现代金融运营模式。数据海量化、多样化、传输快速化和价值化等特征，将给商业银行市场竞争带来全新的挑战 and 机会。

中国银行业现阶段的大数据应用需求大致可以分为以下四类。

- 客户分析：基于各种数据源的客户数据和客户行为数据分析，用于客户分类、客户差异化服务、客户推荐系统、客户流失预测等。
- 风险分析：基于银行交易和客户交互数据进行建模，借助大数据平台快速分析和预测在此发生或者新的市场风险、操作风险等。
- 运营分析：基于企业内外部运营、管理和交互数据分析，借助大数据平台，三百六十度统计和预测企业经营和管理绩效。
- 行业监管：基于企业内外部交易和历史数据，实时或准实时预测和分析欺诈、洗钱等非法行为，遵从法规和监管要求。

10.4.2 大数据时代的银行业发展

大数据的高速发展，使银行业的客户数据、交易数据、管理数据等均呈现爆炸式增长，海量数据席卷而来，机遇和挑战也随之而来，为银行创造变革性价值创造了条件，银行业服务及管理模式都将发生根本性改变。大数据在银行业的应用范围包括客户定价、产品营销、风险管理等（见图10-32）。

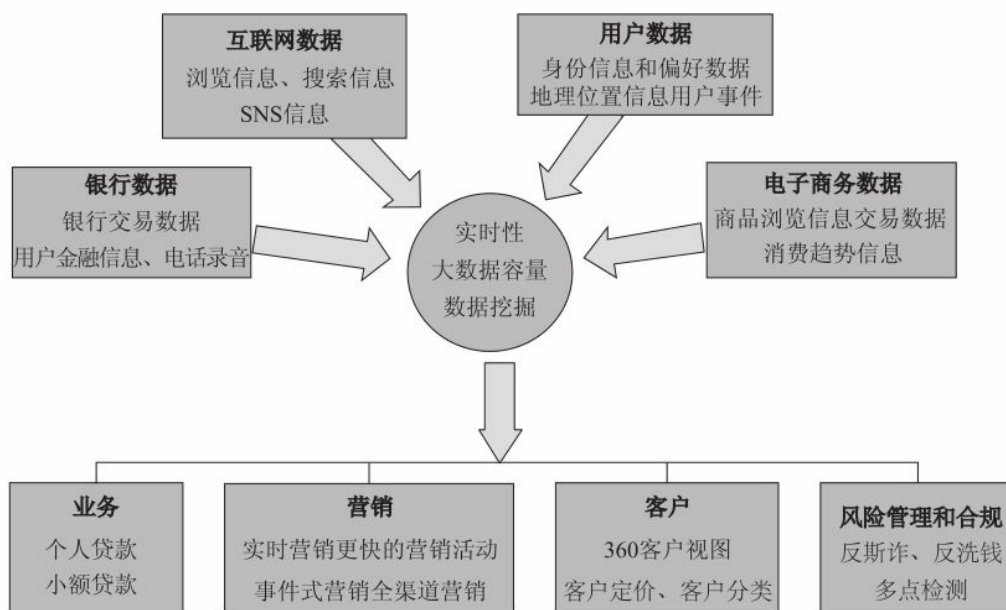


图10-32 中国银行业的大数据应用场景

机遇

大数据时代的银行业发展面临以下机遇。

（1）业务发展空间

我国商业银行所提供的业务服务和产品都具有很强的同质性，但是竞争关系要求银行实施差异化战略。互联网的兴起使得社交媒体成为银行新的接触客户渠道，银行可以从各个网点、PC、移动终端、传感器网络等端口收到结构化、非结构化的海量数据，可深入挖掘客户，强化交叉销售，同时加快产品的创新空间。

（2）决策判断能力

在信息时代，人类社会面临的中心问题将从如何提高生产率转变为如何更好地利用信息来辅助决策。对于银行而言，“大数据”将使银行决策从“经验依赖”向“数据依据”转化，将在深入了解和把握银行自身乃至市场状况的基础上，更加科学地评价经营业绩、评估业务风险、配置全行资源，引导银行业务科学健康发展。

（3）经营管理能力

大数据将掀起银行业的精细化管理革命和竞争。关于资产、负债、客户、交易对手及业务过程中产生的各种数据资产，在风险控制、成本核算、资本管理、绩效考核等方面发挥着重要作用，充分利用数据分析技术将是银行制胜的关键。“数据—信息—商业智能”将逐步成为银行量化、精细化管理的发展路线，为有效提升服务能力提供强大支撑。

挑战

大数据时代的银行发展面临以下挑战。

（1）数据驾驭能力

“大数据”时代首先对银行的数据驾驭能力提出了全新的挑战。在数据收集方面，银行不仅要收集来自网点、信贷等传统渠道的结构化数据，还要收集来自物联网、互联网、机构系统的各类非结构化数据，甚至还要与历史数据对照，非结构化数据收集模式将彻底颠覆银行数据收集理念。在数据存储方面，要达到低成本、低能耗、高可靠性目标，通常要用到冗余配置、分布化和云计算技术，这正是银行所欠缺的。在数据处理方面，有的数据涉及上百个参数，难以用传统的方法描述与度量，处理的复杂度相当大，如客服录音数据等。利用“大数据”的能力将成为决定银行竞争力的关键因素。

（2）生存发展能力

银行的生存发展能力受到挑战。大量的数据来源和强大的数据分析工具正催生出很多新的金融业态来直接瓜分银行的信贷市场。与传统银行相比，互联网金融在信息收集、信息处理、产品交付以及风险防范等方面都有优势，其提供的金融服务已经从简单支付渗透到了转账汇款、小额信贷、现金管理、资产管理、供应链金融、基金和保险代销等银行核心业务领域。

（3）商业运营模式

随着数据化和网络化的全面深入发展，金融服务虚拟化将成为大势所趋。一是产品虚拟化，金融IC卡的推广应用正在逐步提升银行的电子化发展进度，银行资金将越来越多地呈现为各类数据信号的交换，电子货币将与实物货币并驾齐驱。二是服务虚拟化，“善融商务”、“交博汇”以及网络金融商城等银行电子商务平台不断发展，鼠标银行、电子银行成

为未来趋势。三是管理虚拟化，银行业务中的各种单据、凭证等将以数字文件的形式出现，网络成为重要的管理通道，电子化、数据化的管理模式更加方便快捷。传统的商业银行运营模式将逐渐消融在数据化的洪流里，借助“大数据”手段，实现跨越发展，成为未来商业银行可持续发展的唯一选择。

10.4.3 大数据在银行业的发展趋势

目前，大数据相关的技术和工具非常多，给企业提供了更多的选择。在未来，其还会继续出现新的技术和工具，如Hadoop、下一代数据仓库等，这也是大数据领域的创新热点。企业越来越希望能将自己的各类应用程序及基础设施转移到云平台上。就像其他IT系统那样，大数据的分析工具和数据库也将走向云计算。首先云计算为大数据提供了可以弹性扩展、相对便宜的存储空间和计算资源，使得中小企业可以像跨国大企业一样通过云计算来完成大数据分析。其次，云计算IT资源庞大、分布较为广泛，是异构系统较多的企业及时准确处理数据的有力方式，甚至是唯一的方式。随着数据分析集的扩大，以前部门层级的数据集市将不能满足大数据分析的需求，它们将成为企业级数据库的一个子集。

对于银行业来说，“大数据”革命必将颠覆银行传统观念和经营模式。要强化“数据治行”理念，建立分析数据的习惯，重视“大数据”开发利用，提升全行的质量管理、数据管理。同时，其要营造“数据治行”文化，倡导用数据说话，准确描述事实，反映逻辑理性，将现有数据转化为信息资源，为高层管理和决策提供强有力的依据，让决策更加针对目标，让发展更加贴近真实市场。其着眼于“大数据”挖掘和分析，对海量数据的持续实时处理，建设数据仓库项目，为服务质量改善、经营效率提升、服务模式创新提供支撑，全面提升运营管理水平。在项目建设中，其通过梳理、整合经营管理关键数据，建立数据管控体系，搭建基础数据平台。通过数据仓库建设，运用数据挖掘和分析，全方位调整管理模式、产品结构、营销模式、信息战略，从根本上提高风险管理、成本绩效管理、资产负债管理和客户关系管理水平，实现多系统数据的业务逻辑整合，形成全行级客户、产品、协议等主题数据。其积极推动传统业务渠道与移动通信、云计算等新兴业态纵向整合、横向渗透，促进信息集中、整合、共享、挖掘。一方面，要“走出去”，与移动网络、电子商务、社交网络等“大数据平台”完美融合，开展“大数据”分析，为客户提供开放服务平台。另一方面，要“请进来”，与数据分析专业厂商合作，对数据存量进行综合处理与分析。建立完善内容涵盖全面、功能丰富齐

全，集网上贸易服务、网上保理、电子商业汇票、票据池、应收账款池融资、在线融资等为一体的综合供应链金融服务体系，为客户提供触手可及的全方位贴身服务。

10.4.4 大数据在金融行业的实践

问题和需求

随着互联网金融的快速兴起，银行业务竞争激烈，A金融机构急需以金融大数据查询、分析、挖掘为基础，进行产品创新、预测和风险评估，改善服务质量，提升竞争力。但A金融机构的IT数据处理平台架构还是以传统数据库与数据仓库为主的模式（见图10-33）。

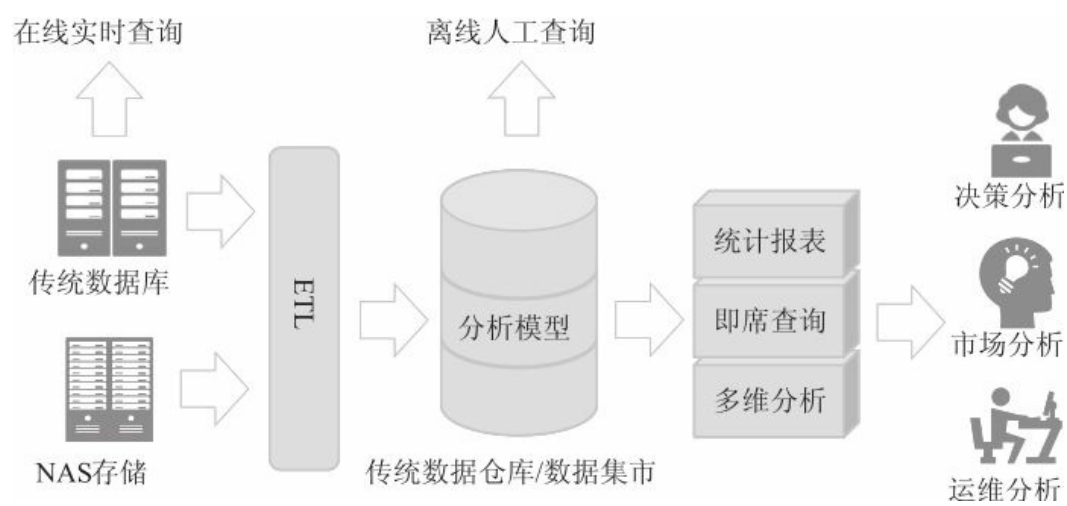


图10-33 以传统数据库与数据仓库为主的银行数据处理平台

传统面向结构化数据的数据平台已经无法满足大数据数量、种类的快速增长，无法支撑A金融机构未来对数据处理的要求。具体表现在以下几个方面。

- 系统无法支撑在线查询5年历史明细：业务需要在线查询5年15TB的交易历史明细，但现状是容量受限，数据库仅能在线查询1年3TB的交易历史明细，1年以上历史明细需要数据仓库离线人工查询，客户等待时间长。
- 系统无法支撑全量多维客户行为分析：业务需要全量多维度分析，改进分析机制，提高分析结果转化率，但现状是受限于处理能

力制约，ETL抽取到传统数据库的数据维度较少，且现有SAS分析工具采用专家经验机制，导致分析结果转化率低。

- 系统无法支撑实时征信：征信数据分散在业务数据库、数据仓库，信用卡开户、透支服务征信数据需要分时分散查询，且流程人工干预多，导致业务办理需要3周左右，无法实现实时征信数据查询客户满意度低。

- 数据库问题：原有数据系统仅适合实时交易类业务，不适合分析类业务，存储结构化数据，容量有限，无法存储过多的历史数据。

- 数据仓库问题：原有数据仓库系统仅适合结构化数据分析类业务，非结构化数据需前端ETL工具抽取，转换成结构化数据存储，但数据维度减少，无法线性扩展，数据量大处理缓慢，银行一般多业务分时复用使用，无法多业务并发使用。

基于以上诸多挑战和问题，A金融机构寻求采用大数据技术来完善金融数据处理解决方案，来应对日益突出的业务问题。在大数据解决方案选择上，A金融机构对大数据平台有三个核心的要求。

- 安全性：满足金融等要求；
- 可靠性：所有组件支持HA，支持容灾；
- 易用性：与应用无缝衔接，适合现有编程习惯。

架构方案及优点

A金融机构对多家知名IT厂商的大数据解决方案进行了详细的考察和测试验证，搭建了一个全新的金融大数据处理平台，其架构如图10-34所示。

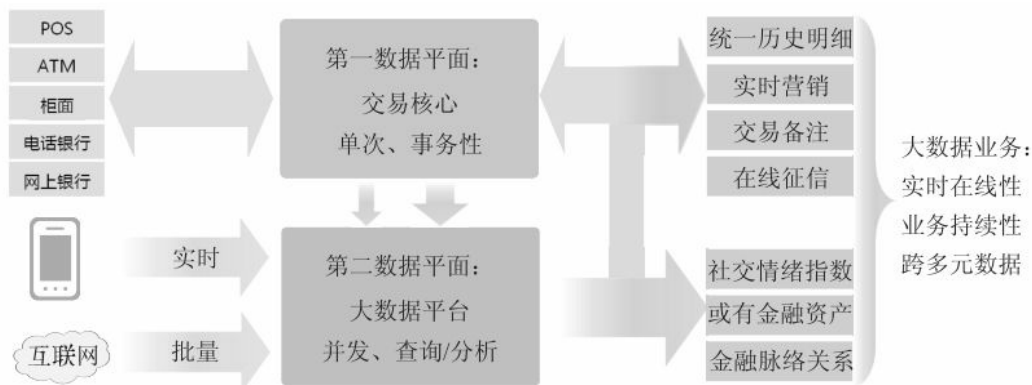


图10-34 A金融机构全新的金融数据架构

在这个架构中，A金融机构将数据分为第一数据平面和第二数据平面。第一数据平面主要基于原有的金融IT平台，以交易为核心，支撑传统的金融数据处理与分析业务。第二数据平面则是以大数据平台为核心的新建数据平面，主要处理金融数据分析业务，实现诸如社交情绪指数、或有金融资产、金融脉络关系、在线征信、精准推荐、在线历史明细等业务。第一平面与第二平面的数据实时共享与交互，实现相互间业务支撑。

该架构除了实现基于Hadoop的基础大数据分析能力外，还实现了A金融机构所需的安全、可靠、易用三大特性。

- **高安全性：**业界第一家支持金融等级保护、第一家支持RBAC用户组权限管理和消除HDFS明文存放隐患的大数据平台。
- **高可靠性：**大数据平台全组件支持HA，业界第一个通过1000+KM异地容灾验证的厂家。
- **易用性：**丰富的行业全量建模和二次开发能力，让大数据与客户应用无缝衔接，全自动化在线运维、自动化的应用开发助手，轻松管理大数据系统。

在线历史明细查询解决方案中，由IT供应商提供完整的大数据分布式业务平台和Hadoop大数据平台解决方案，A金融机构的技术团队只需专注历史明细查询业务的编写。分布式业务平台支持多业务系统并发访问，实现实时历史明细查询能力。方案同时支持Socket、Web业务请求接入和分发，与A金融机构业务系统无缝衔接。创新的CTBase方案、独有的

表聚簇和多级索引支持HBase多表关联查询的能力。HBase同时支持SQL、Java API编程接口，适应客户的编程习惯。

在全量多维客户行为分析解决方案中，由IT供应商提供完整的数据挖掘工具大数据洞察平台和Hadoop大数据平台解决方案，A金融机构技术团队只需专注客户行为分析业务的编写。IT供应商同时提供了金融通用的客户行为分析业务，即用户特征刻画、小微贷倾向分析。大数据洞察平台基于大数据全量建模分析，可以挖掘出14 000位客户特征，实现多维并发分析。方案采用Hadoop机器自动学习机制，大大提高分析准确度。客户行为分析结果存储在HBase，供业务查询使用。

在实时征信解决方案中，IT供应商提供完整的大数据分布式业务平台（DAP）和Hadoop大数据平台解决方案，A金融机构技术团队只需专注实时征信业务组件的编写。分布式业务平台提供征信流程业务系统并发处理能力。分布式业务平台提供实时工作流引擎，可以根据预先设置的顺序自动化执行征信的每一个业务流程。HBase中存储所有与实时征信相关的表，包括资料信息表、内部资信表、客户公共profile表、风险信息表、征信主表，提供征信数据实时查询能力。HBase支持SQL调用接口，与实时征信已有的业务组件无缝衔接。

系统收益

A金融机构建设的第二数据平面大数据平台为A金融机构带来非常显著的价值收益，具体表现在以下几个方面（见图10-35）。

- 历史明细查询：实现统一集中存储5年15TB交易历史明细数据，便于管理和扩展，多业务系统并发实时查询5年交易历史明细数据，提升金融最终用户的使用体验。
- 实时征信：实现了2~5秒自动化征信，提升金融信贷客户的满意度；实时工作流引擎简化客户部署实时自动化征信业务的难度。
- 客户行为分析：小微贷倾向分析实现TOP10 000客户推荐成功转化率，相比传统数据仓库，四维度分析模式转化率将提高6倍。

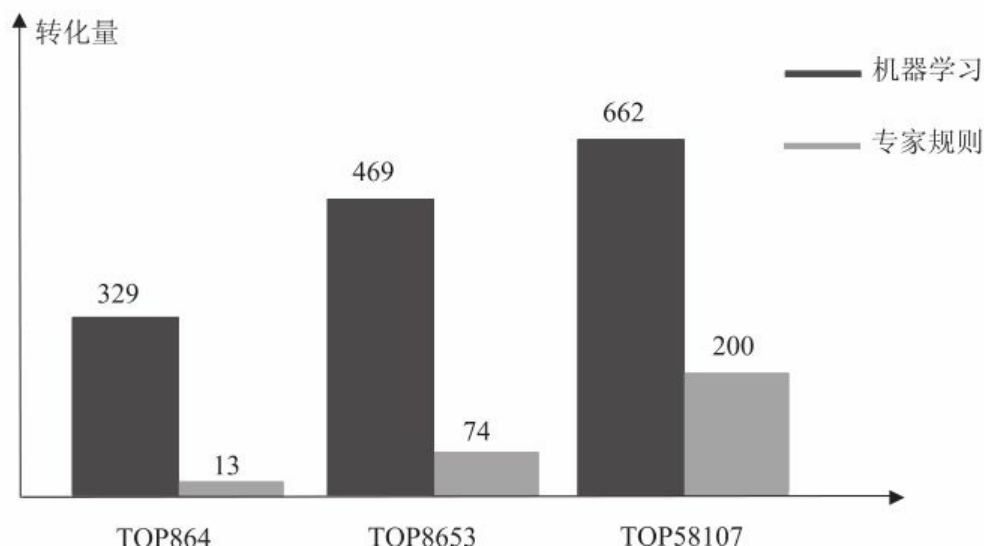


图10-35 小微贷大数据与传统数据仓库分析效果对比

10.5 未来大数据应用畅想

10.5.1 身边的大数据

似乎一夜之间，大数据（Big Data）变成一个IT行业中最时髦的词汇。

首先，大数据不是什么完完全全的新生事物，Google的搜索服务就是一个典型的大数据运用，根据客户的需求，Google实时从全球海量的数字资产（或数字垃圾）中快速找出最可能的答案，并呈现给你，这就是一个最典型的大数据服务。只不过过去这样规模的数据量处理和有商业价值的应用太少，在IT行业没有形成成型的概念。现在随着全球数字化、网络宽带化、互联网应用于各行各业，累积的数据量越来越大，越来越多企业、行业和国家发现，可以利用类似的技术更好地服务客户、发现新商业机会、扩大新市场以及提升效率，从而逐步形成大数据这个概念。

从运营商行业的一个真实故事讲起：某运营商客户有一个困扰，即一些手机厂商和客户内部的某些人内外勾结，采用俗称“洗码”的方式来非法获利，具体来说，就是这些手机厂商的部分手机进入运营商渠道销售、加上资费包出库后，并没有进入最终消费者手中，而是进入相关利益者手中，相关利益者把手机资费拆包卖掉后，手机又通过渠道重新进入运营商的销售环节。对于运营商来说，这样没有发展新用户，只是循环徒

增成本。单纯从运营商的流程和业务系统是找不出问题在哪里的。后来，客户技术团队利用技术手段从网络、运营系统等抓取了很多的数据，对各品牌、各款式、各批次手机建模分析，发现一系列线索，找到了问题点和相关责任人，运营商通过处罚及威慑基本杜绝了这种情况的发生。这件事当时做的时候，大数据概念还没有变得很热，但毫无疑问，这样的做法就是大数据方法的有效实践。

另外一个有趣的故事是关于奢侈品营销的。PRADA在纽约的旗舰店中每件衣服上都有RFID码。每当一个顾客拿起一件PRADA进试衣间，RFID会被自动识别。同时，数据会传至PRADA总部。每一件衣服在哪个城市哪个旗舰店什么时间被拿进试衣间停留多长时间，数据都被存储起来加以分析。如果有一件衣服销量很低，以往的做法是直接下架。但如果RFID传回的数据显示这件衣服虽然销量低，但进试衣间的次数多，那就能另外说明一些问题。也许这件衣服的下场就会截然不同，也许在某个细节的微小改变就会重新创造出一件非常流行的产品。

还有一个是关于中国粮食统计的故事。中国的粮食统计是一个老大难的问题。中国的统计，虽然有组织、有流程、有法律，但中央的统计人员依靠省统计人员，省靠市，市靠县，县靠镇，镇靠村，最后真正干活或上报的是基层兼职的调查人员。在前两年北京的一个会议上，原国家统计局总经济师姚景源讲述了他们是如何提升数据的精准性的。他们采用遥感卫星，通过图像识别，把中国所有的耕地标识计算出来，然后把中国的耕地网格化，对每个网格的耕地抽样进行跟踪、调查和统计，然后按照统计学的原理，计算（或者说估算）出中国整体的粮食数据。这种做法是典型采用大数据建模的方法，打破传统流程和组织，直接获得最终的结果，或者获得不同渠道来源的数据，对同一个事实从不同侧面加以验证。

最后是一个炒股的故事。这个故事来自2011年好莱坞的一部高智商电影《永无止境》，讲述一位落魄的作家库珀，服用了一种可以迅速提升智力的神奇蓝色药物，然后他将这种高智商用于炒股。库珀是怎样炒股的呢？他能在短时间掌握无数公司的资料和背景，也就是将世界上已经存在的海量数据（包括公司财报、电视、几十年前的报纸、互联网、小道消息等）挖掘出来，串联起来，通过海量信息的挖掘、分析，使一切内幕都不是内幕，使一切趋势都在眼前，结果在10天内他就赢得了200万美元，神奇的表现让身边的职业投资者目瞪口呆。这部电影展现了大数据魔力，我们推荐没有看过的IT人士看一看。

从这些案例来看，大数据并不是很神奇的事情。就如同电影《永无止境》提出的问题：人类通常只使用了20%的大脑，如果剩余80%的大脑潜能被激发出来，世界会变得怎样？在企业、行业和国家的管理中，通常只有效使用了不到20%的数据（甚至更少），如果剩余80%的数据价值激发起来，世界会变得怎么样呢？特别是随着数据爆发式增长，并且得到更有效应用，世界会怎么样呢？

单个数据并没有价值，但越来越多的数据累加，量变就会引起质变，就好像一个人的意见并不重要，但1千人、1万人的意见就比较重要，上百万人就足以掀起巨大的波澜，上亿人足以改变一切。

数据未被整合和挖掘，其价值无法呈现出来。《永无止境》中的库珀如果不能把海量信息围绕某个公司的股价整合起来、串联起来，这些信息就没有价值。

因此，海量数据的产生、获取、挖掘及整合，使之展现出巨大的商业价值，数据驱动经营，这就是我们所理解的大数据。在互联网对一切重构的今天，这些问题都不是问题。因为，大数据是互联网深入发展的下一波应用，是互联网发展的自然延伸。目前，可以说大数据的发展到了一个临界点，才会成为IT行业中最热门的词汇之一。

10.5.2 大数据将重构很多行业的商业思维和商业模式

我们以对未来汽车行业的狂野想象来感受大数据可能带来的变化。

在人的一生中，汽车是一项巨大的投资。以一部30万元的车、7年换车周期来算，每年折旧费4万多元（这里还不算资金成本），加上停车、保险、油、维修、保养等各项费用，每年花费应在6万元左右。汽车产业也是一个拥有很长产业链的龙头产业，这个方面只有房地产可以媲美。

但同时，汽车产业链是一个低效率、变化慢的产业。汽车一直以来就是四个轮子、一个方向盘、两排沙发（李书福语）。这么一个昂贵的东西，围绕车产生的数据却少得可怜，行业产业链之间几乎无任何数据传递。

我们在这里狂野地想象一番，如果将汽车全面数字化，都大数据了，会产生什么结果？

有些人说，汽车数字化，不就是加个MBB模块吗？不，这太小儿科了。深入地来看，数字化意味着汽车可以随时连上互联网，意味着汽车是一个大型计算系统加上传统的轮子、方向盘和沙发，意味着可以数字化导航、自动驾驶，意味着你和汽车相关的每一个行动都数字化，包括每一次维修、每一次驾驶路线、每一次事故的录像、每一天汽车关键部件的状态，甚至你的每一个驾驶习惯（如每一次的刹车和加速）都记录在案。这样，你的车每月甚至每周都可能产生1TB以上的数据。

好了，我们假设这些数据都可以存储并分享给相关的政府、行业和企业。这里不讨论隐私问题带来的影响，假设在隐私保护的前提下，数据可以自由分享。

那么，保险公司会怎么做呢？保险公司把你的所有数据拿过去建模分析，发现几个重要的事实：一是你开车主要只是上下班，A地到B地这条线路是非繁华路线，红绿灯很少，这条路线过去一年统计的事故率很低；你的车况（车的使用年限、车型）好，此车型在全市也是车祸率发生较低的车型；甚至可以统计你的驾驶习惯，加油平均，临时刹车少，超车少，和周围车保持了应有的车距，驾驶习惯好。最后结论是，你的车型好，车况好，驾驶习惯好，常走的线路事故率低，过去一年也没有出过车祸，因此可以给予更大幅度的优惠折扣。这样保险公司就完全重构了它的商业模式了。在没有大数据支撑之前，保险公司只把车险客户做了简单的分类，一共分为四种客户，第一种是连续两年没有出车祸的，第二种是过去一年没有出车祸的，第三种是过去一年出了一次车祸的，第四种是过去一年出了两次及以上车祸的。这种简单粗略的分类，就好像女人找老公，仅把男人分为没有结过婚的、结过一次婚的、结过二次婚的、结过三次及以上婚的四种男人，就敢嫁人一样。在大数据的支持下，保险公司可以真正以客户为中心，把客户分为成千上万种，每个客户都有个性化的解决方案，这样保险公司的经营将完全不同，对于风险低的客户敢于大胆地给予折扣，对于风险高的客户报高价甚至拒绝。如此一来，一般的保险公司将难以和这样的保险公司竞争。拥有大数据并使用大数据的保险公司比传统公司将拥有压倒性的竞争优势，大数据将成为保险公司最核心的竞争力，因为保险就是一个基于概率评估的生意，对于准确评估概率，大数据毫无疑问是最有利的武器，而且简直是量身定做的武器。

在大数据的支持下，4S店的服务也将完全不同。车况信息会定期传递到4S店，4S店会根据情况及时提醒车主及时保养和维修，特别是对于可能

危及安全的问题，在客户同意下甚至会采取远程干预措施，同时还可以提前备货，车主一到4S店就可以维修而不用等待。

对于驾驶者来说，不想开车的时候，在大数据和人工智能的支持下，车辆可以自动驾驶，并且对于你经常开的线路可以自学习、自优化。

Google的自动驾驶汽车，为了对周围环境做出预测，每秒钟要收集差不多1GB的数据，没有大数据的支持，自动驾驶是不可想象的；在与周围车辆距离过近的时候，会及时提醒车主避让；上下班的时候，会根据实时大数据情况，对于你经常开车的线路予以提醒，绕开拥堵点，帮你选择最合适的线路；在出现紧急状况的时候，比如爆胎，自动驾驶系统将自动接管，提高安全性（人一辈子可能难以碰到一次爆胎，人在紧急时的反应往往是灾难性的，只会更糟）；到城市中心，寻找车位是一件很麻烦的事情，但未来你可以到了商场门口后，让汽车自己去找停车位，等想要回程的时候，提前通知汽车，让汽车自己开过来接你。

车辆是城市最大、最活跃的移动物体，是拥堵的来源，也是最大的污染源之一。数字化的车辆、大数据应用将带来很多的改变。红绿灯可以自动优化，根据不同道路的拥堵情况自动进行调整，甚至在很多地方可以取消红绿灯；城市停车场也可以大幅度优化，根据大数据的情况优化城市停车位的设计，如果配合车辆的自动驾驶功能，停车场可以进行革命性的演变，设计专门为自动驾驶车辆服务的停车楼，地下、地上楼层可以高达几十层，停车楼层可以更矮，只要高于车高度即可（或者把车竖起来停），这样将会对城市规划产生巨大的影响；在出现紧急情况时，如前方塌方的时候，可以第一时间通知周围车辆（尤其是开往塌方道路的车辆）；现在的燃油税也可以发生革命性变化，可以真正根据车辆的行驶路程，甚至根据汽车的排污量来收费，排污量少的车甚至可以搞碳交易，将排放量卖给高油耗的车；政府还可以每年公布各类车型的实际排污量、税款、安全性等指标，鼓励民众买更节能、更安全的车。

电子商务和快递业也可能发生巨大的变化。运快递的车可以自动驾驶，不用在白天行驶在拥堵的道路，而在晚上行驶。你可以在家门口设计自动接收箱，通过开启密码自动投递进去，就好像过去报童投报一样。

这么想象下来，我们看到，汽车数字化、互联网化、大数据应用、人工智能等将对汽车业及相关的长产业链产生难以想象的巨大变化和产业革命，具有无限的想象空间，可能完全被重构。当然，要实现我们所描述的场景，还有很长的路要走。

下面，我们总结一下大数据将会带来的改变。

第一，大数据使企业真正有能力从以自我为中心改变为以客户为中心。企业是为客户而生的，目的是为股东获得利润。只有服务好客户，才能获得利润。但过去，很多企业是没有能力做到以客户为中心的，原因就是相应客户的信息量不大，挖掘不够，系统也不支持，目前的保险业就是一个典型。大数据的使用能够使对企业的经营对象从客户的粗略归纳（提炼归纳的“客户群”）还原成一个个活生生的客户，这样经营就有针对性，对客户的服务就更好，投资效率就更高。

第二，大数据在一定程度上将颠覆了企业的传统管理方式。现代企业的管理方式来源于对军队的模仿，依赖于层层级级的组织和严格的流程，依赖信息的层层汇集、收敛来制定正确的决策，再通过决策在组织的传递与分解以及流程的规范，确保决策得到贯彻，确保每一次经营活动都有质量保证，也确保一定程度上对风险的规避。过去这是一种有用而笨拙的方式。在大数据时代，我们可能重构企业的管理方式，通过大数据的分析与挖掘，大量的业务本身就可以自决策，不必要依靠庞大的组织和复杂的流程。大家都是基于大数据来决策，都是依赖于既定的规则来决策，是高高在上的首席执行官决策，还是一线人员决策，本身并无大的区别，那么企业是否还需要如此多层级的组织和复杂的流程呢？

第三，大数据另外一个重大的作用是改变了商业逻辑，提供了从其他视角直达答案的可能性。现在人的思考或者企业的决策，事实上都是逻辑的力量在主导起作用。我们去调研、收集数据、进行归纳总结，最后形成自己的推断和决策意见，这是一个观察、思考、推理、决策的商业逻辑过程。人和组织的逻辑形成需要大量的学习、培训与实践，代价是非常巨大的。但这是否是唯一的道路呢？大数据给了我们其他的选择，就是利用数据的力量，直接获得答案。就好像我们学习数学，小时候学九九乘法表，中学学几何，大学学微积分，碰到一道难题，我们是利用了多年学习沉淀的经验来努力求解，但我们还有一种方法，在网上直接搜索是不是有这样的题目，如果有，直接抄答案就好了。很多人就会批评说，这是抄袭，是作弊。但我们为什么要学习啊？不就是为了解决问题嘛。如果我任何时候都可以搜索到答案，可以用最省力的方法找到最佳答案，这样的搜索难道不可以是一条光明大道吗？换句话说，为了得到“是什么”，我们不一定要理解“为什么”。我们不是否定逻辑的力量，但是至少我们有一种新的巨大力量可以依赖，这就是未来大数据的力量。

第四，通过大数据，我们将以全新的视角来发现新的商业机会和重构新的商业模式。我们现在看这个世界，比如分析家中食品腐败，主要就是依赖于我们的眼睛再加上我们的经验，但如果我们有一台显微镜，我们一下就看到坏细菌，那么分析起来就完全不一样了。大数据就是我们的显微镜，它可以让我们从全新视角来发现新的商业机会，并可能重构商业模式。我们的产品设计可能不一样了，很多事情不用猜了，客户的习惯和偏好一目了然，我们的设计能轻易命中客户的心窝；我们的营销也完全不同了，我们知道客户喜欢什么、讨厌什么，更有针对性。特别是显微镜再加上广角镜，我们将有更多全新的视野了。这个广角镜就是跨行业的数据流动，使我们过去看不到的东西都能看到了，比如前面所述的汽车案例，开车是开车，保险是保险，本来不相关，但当我们把开车的大数据传递到保险公司以后，整个保险公司的商业模式就完全重构了。

第11章 企业私有云和公有云对IAAS层的诉求

11.1 企业私有云和公有云对IAAS层的诉求

随着云计算技术、产品、解决方案的发展，企业私有云建设和公有云计算均进入新一轮建设高潮，同时也面临着新的问题。

多厂家产品集成带来的建设周期长的问题

私有云和公有云建设需要购买服务器、存储、网络、操作系统、数据库、虚拟化软件、云基础设施软件、管理软件，最终构建成企业所需要的私有云基础设施。在规划和采购过程中，客户根据自己的经验，挑选符合建设目标的设备，因此在建设过程中，面临多个设备厂家、多种设备型号、多种软件资源（虚拟化软件、操作系统、数据库、双机等）的异构集成和整合，例如服务器和存储的集成，网络设备和计算/存储设备的对接；同时客户也必须完成软件和硬件的对接，例如OS同应用软件和硬件的集成，与新的管理软件的集成，需要全面的软硬件对接联调保证云基础设施可以正常工作。客户往往需要借助技术理解深、协调能力强、组织支撑到位的集成商才能完成私有云或公有云的建设，不仅需要投入大量的集成服务成本，而且会面临云基础设施建设周期长，无法有效支撑业务发展等问题。

业务和应用迁移

私有云和公有云的建设是为了支持企业的IT业务发展，即需要承载企业的IT业务/应用。因此，快速、平滑地将原有IT业务和应用迁移到新的基础设施上，或部署新的业务/应用是云基础设施建设过程中的关键任务。首先，企业必须确定哪些业务/应用可以采用物理部署或虚拟化部署，不是所有的业务都适合采用虚拟化部署，也不是所有的应用都可以迁移到虚拟化平台上，因此企业必须了解和确定迁移策略和方案。其次，当业务/应用迁移时，在云基础设施上，各类应用需要分配何种类

型资源，各设备之间需要如何配合，如何保证各业务的可靠性、安全隔离，是企业必须考虑的第二个问题。第三，应用迁移后，如何对应用的性能进行调优，保证应用SLA，快速定位和解决问题，是企业必须面对的第三个问题。原有业务/应用的迁移，新业务/应用的快速部署，是决定云基础设施成败的关键。

多厂家设备管理难度

多厂家设备不仅仅需要IT维护人员需要具备多种设备的维护能力，对人力成本和人员培训提出了更高的要求。多厂家设备也为快速问题定位带来巨大的挑战。在云计算环境下，资源高度的虚拟化，应用程序与组件是多对多的依赖关系，一个应用程序出了故障，很难快速、准确地定位是哪个厂家的哪个组件出现了问题，因此，问题故障的定位是云计算环境下数据中心管理的又一大挑战。

系统平滑扩展问题

随着BYOD，移动、社交、大数据的发展，企业需要能够快速扩容云基础设施为业务和员工提供更多的IT服务，因此，无论是企业云还是公有云，都需要有平滑扩展能力。首先需要快速扩展系统，提供更多的资源；其次，扩展时不能影响现有业务；第三，扩展后的系统要能够充分利用扩容前的资源和扩容后的资源，提升资源利用率。

为了解决客户在私有云和公有云建设过程中面临的挑战，云计算设备供应商开始纷纷推出一体机设备，HP在2007年推出业界首款一体机CloudSystem，之后业界开始涌现出数据库一体机、数据仓库一体机、基础设施一体机、参考架构一体机、计算存储融合一体机等适用于各种业务场景的一体机。2012年一体机总市场超过30亿美金，并在未来的5年以40%的年增长率快速增长，这标志着客户对一体机的态度已经从尝试、认可到大规模采购的转变。一体机已经成为云基础设施的关键成员。

目前，业界对于一体机（Appliance）并没有一个通用的定义，从一体机应用场景和客户对一体机的需求角度来分析，一体机应该具备的通用特征：

简化的基础设施组件

客户需要能够快速部署、易于维护、快速定位问题的基础设施组件，因此，一体机系统是结构简洁、接口统一、各组件的管理系统统一的基础设施。例如有一套统一的管理系统，柜内通信部需要外置交换机等。

融合的基础设施组件

传统的计算、存储、网络三大件模型是为了各厂家设备对接的水平划分架构，当一体机作为基础设施建设基本组件时，本身可看成是一个提供融合了计算、存储、网络资源的实体，因此，一体机网络可以采用各种融合的技术，例如，计算和存储融合，由服务器提供计算资源和存储资源，不需要外置的存储；计算和网络的融合，机框内交换板提供框内和框间交换功能，不需要外置交换机。融合技术在一体机的应用即可以简化一体机结构，也可提升性能，并降低成本，是一体机的关键技术之一。

企业IT业务发展要求IT基础设施可以快速、平滑扩展，一体机厂家采用各种技术来支持平滑扩展，例如计算和存储融合架构的一体机，直接增加服务器就可以扩展计算和存储资源，不仅简单、快速而且成本低。

11.2 一体机的市场和技术

11.2.1 一体机市场

自从HP在2007年推出业界首款一体机CloudSystem，Oracle在2008年推出软硬件集成的数据库云服务器Oracle Exadata，一体机被业界广泛接受，顺应企业需求，业界已经推出三类一体机产品。

（1）应用一体机：专门承载某类应用或功能的一体机，如Oracle Exadata一体机，SAP HANA一体机，大数据一体机。

）

（2）基础设施一体机：集成了服务器，存储设备，网络设备的开机即用的基础设施设备，厂家通常提供多种型号来满足各种规模企业客户的需要，典型产品如Vblock 100/200/320/720等。

（3）参考架构一体机：提供一个基础设施设备所需的服务器，存储设

备，网络设备的型号和规格，客户可以按需选择部件型号和规格来拼装一体机，典型产品如Flexpod。

应用一體機的主要供應商為應用軟件廠商，例如應用一體機中排名第一的Exadata，其優勢在於業界份額第一的Oracle數據庫和數據倉庫軟件。

基礎設施一體機和參考架構一體機的供應商以硬件廠家為主，廠家基於其硬件產品對客戶的影響力和對客戶需求的深入把握，主流廠家包括EMC、CISCO、HP、IBM。

自一體機推出後，一體機銷售快速增長。據統計2012年一體機的總收入為約40億美元，較2011年增長50%，三類一體機收入基本相當，約各占1/3。據IDC預測，2016年一體機市場空間將達到178億美元。

目前一體機的主要市場在美國，歐洲和亞太區域。北美和歐洲的工業化水平較高，IT基礎設施需求一直比較強勁，同時由於人力成本較高，因此市場對於部署快速，維護簡單的一體機持接受態度，亞太地區由於中國等經濟發展較快國家對於IT設備需求旺盛，同時由於IT維護人才經驗等因素，對方便擴展，維護簡單的一體機也接納很快。據統計，2012年一體機銷售中北美市場約50%，歐洲和亞太各約25%。

業界通常將IT基礎設施市場分成四類。

- （1）中小企業：通常簡稱為SMB，這類企業的IT設施規模不大。
- （2）大企業：企業的IT設施規模較大，IT應用規模和種類都較多。
- （3）服務提供商：主要是指向企業或最終客戶提供電信業務、互聯網業務、數據中心外包等業務的提供商。
- （4）超大规模業務提供商：主要是指一些全球性的業務提供商，例如搜索巨頭Google，電子商務兼雲計算服務提供商Amazon，這些全球性IT業務巨頭具有大量IT研發工程師和維護工程師，自己開發業務系統，因此往往需要定制硬件，通常不會採用一體機。

目前一體機客戶集中於中小企業，大企業和部分服務提供商，由於小規模、低成本一體機產品比較少，因此主要客戶來自大企業。

11.2.2 一体机技术

NoSQL/MapReduce技术

随着IT技术的发展，信息处理的对象发生了极大的变化，从关系数据的处理到非关系数据的处理方向转变，从中小规模数据量（TB）处理到大规模数据（PB/ZB）处理转变。与此呼应，新数据处理技术不断涌现，其中典型的是非关系型数据库技术NoSQL和分布式处理技术MapReduce。

NoSQL

NoSQL是Not only SQL的缩写，是对不同于传统的关系型数据库的数据库管理系统的统称。

两者存在许多显著的不同点，其中最重要的是NoSQL不使用SQL作为查询语言。其数据存储可以不需要固定的表格模式，也经常会避免使用SQL的JOIN操作，一般有水平可扩展性的特征。NoSQL的实现具有两个特征：使用硬盘，或者把随机存储器作存储载体。

NoSQL一词最早出现于1998年，是Carlo Strozzi开发的一个轻量、开源、不提供SQL功能的关系数据库。

2009年，Last.fm的Johan Oskarsson发起了一次关于分布式开源数据库的讨论，来自Rackspace的Eric Evans再次提出了NoSQL的概念，这时的NoSQL主要指非关系型、分布式、不提供ACID的数据库设计模式。

2009年在亚特兰大举行的“no:sql（east）”讨论会是一个里程碑，其口号是“SELECT fun, profit FROM real_world WHERE relational=false”。因此，对NoSQL最普遍的解释是“非关联型的”，强调Key-Value Stores和文档数据库的优点，而不是单纯地反对RDBMS。

当代典型的关系型数据库在一些应用中表现出糟糕的性能，例如为巨量文档建立索引、高流量网站的网页服务，以及发送流式媒体。关系型数据库的典型实现主要被调整用于执行规模小而读写频繁，或者大批量极少写访问的事务。

MapReduce

MapReduce是Google提出的一个软件架构，用于大规模数据集（大于1TB）的并行运算。概念“Map（映射）”和“Reduce（化简）”，及他们的主要思想，都是从函数式编程语言借来的，还有从矢量编程语言借来的特性。

当前的软件实现是指定一个Map（映射）函数，用来把一组键值对映射成一组新的键值对，指定并发的Reduce（化简）函数，用来保证所有映射的键值对中的每一个共享相同的键组。

简单来说，一个映射函数就是对一些独立元素组成的概念上的列表（例如，一个测试成绩的列表）的每一个元素进行指定的操作（比如，有人发现所有学生的成绩都被高估了一分，他可以定义一个“减一”的映射函数，用来修正这个错误）。事实上，每个元素都是被独立操作的，而原始列表没有被更改，因为这里创建了一个新的列表来保存新的答案。这就是说，Map操作是可以高度并行的，这对高性能要求的应用以及并行计算领域的需求非常有用。

化简操作指的是对一个列表的元素进行适当的合并。继续看前面的例子，如果有人想知道班级的平均分该怎么做？他可以定义一个化简函数，通过让列表中的奇数（odd）或偶数（even）元素跟自己的相邻的元素相加的方式把列表减半，如此递归运算直到列表只剩下一个元素，然后用这个元素除以人数，就得到了平均分。虽然它不如映射函数那么并行，但是因为化简总是有一个简单的答案，大规模的运算相对独立，所以化简函数在高度并行环境下也很有用。

SSD技术

SSD（Solid-State Drive）固态硬盘，采用NAND Flash颗粒，它能够有效填补在CPU计算能力越来越强的情况下，内存与HDD之间越来越大的IOPS和时延的缺口。

PCIe SSD提供高达600K@4KB 100% Random读IOPS性能以及3000MBPS读写带宽，满足客户对高I/O、高带宽的业务需求（见图11-1、图11-2）。

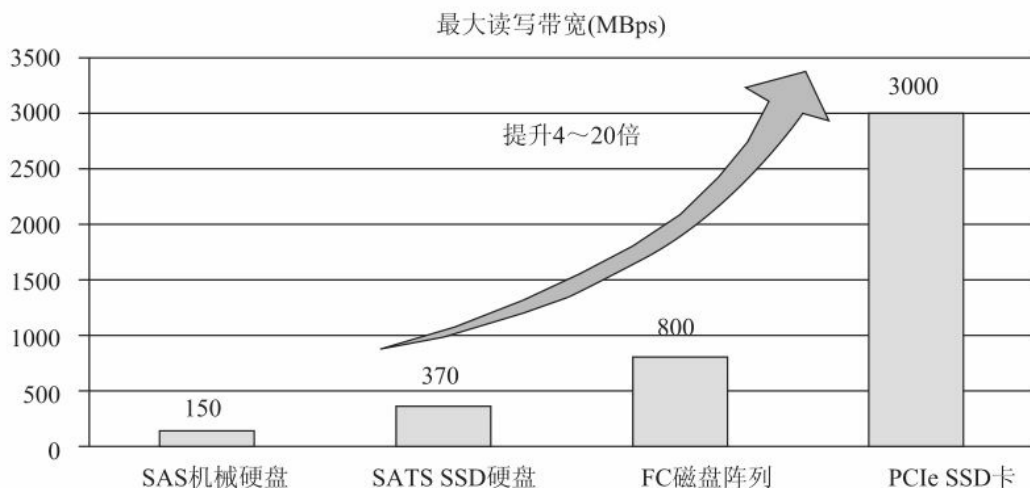


图11-1 PCIe SSD与磁盘、SSD盘、FC阵列读写带宽对比

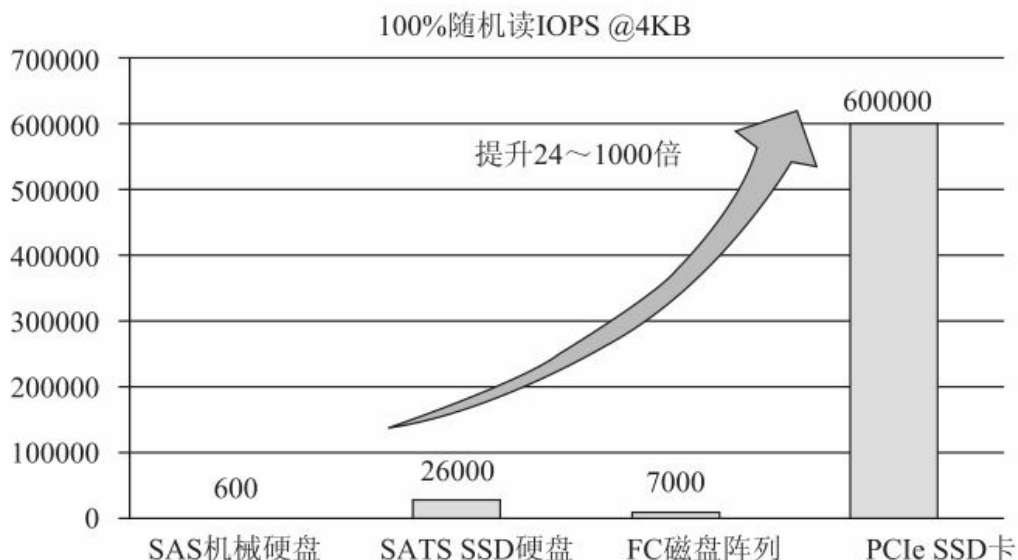


图11-2 PCIe SSD与磁盘、SSD盘、FC阵列IOPS对比

SSD已经被广泛应用，不仅被用做Cache缓存，而且在数据库、高性能存储等领域，SSD还被作为主存替代传统的磁盘，以获得更好的IOPS和读写时延。如Oracle的数据库一体机Exadata、VCE的Vblock/Flexpod、Nutanix的基础设施一体机、华为的FusionCube基础设施一体机等（见图11-3）。

NAND Flash分为三种，分别是单层式存储（SLC）、多层式存储（MLC，通常用来指称两层式存储）、三层式存储（TLC）。随着制造工艺的不断改进，成本持续下降，但是SLC的成本仍然过高，MLC被大

量用于企业应用。

SSD技术的关键挑战是寿命和制造工艺的提升。随着制造工艺的提升，每单元容纳的电子数量越来越少（30@20nm），2D NAND Flash在12/10nm遭遇天花板，3D技术可以缓解可靠性瓶颈，但无法根本解决，3D NAND Flash技术越来越成熟，在2014—2015年量产。随着工艺提升到1xnm，对控制器算法纠错的能力要求越来越高，控制器将走向ASIC化。

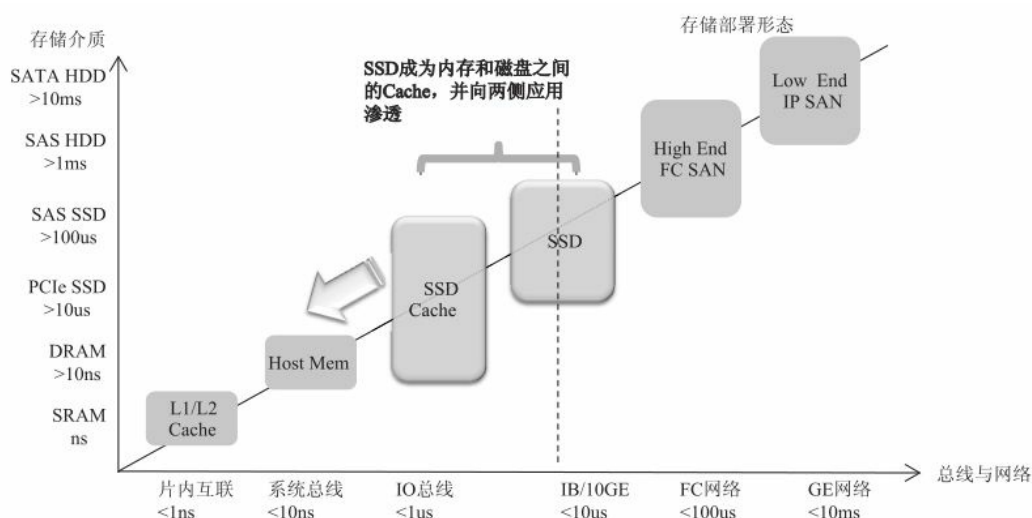


图11-3 SSD成为内存和磁盘之间的Cache，并向两侧应用渗透

SSD发展轨迹具体如下：

- 接口不断丰富，速率不断提升，和IT业界接口互连技术发展的整体节奏保持一致；
- 控制器ASIC化，且商业ASIC控制器逐步被少数厂商把控，部分系统厂商选择自研SSD控制器；
- 针对NAND FLASH介质的可靠性/高性能优化技术在不断深入，特别是系统应用的改进。

NAND FLASH发展轨迹具体如下：

- 工艺不断进步，且发展呈加速趋势，3D NAND预计在2015—

2016年会走向批量商用；

- 纠错能力要求越来越高，控制器实现复杂度增加；
- 单Die容量越来越大，单位容量成本不断降低，逐步逼近企业级SAS HDD。

NVRAM和PCIe SSD性能规格对比，如图11-4所示。

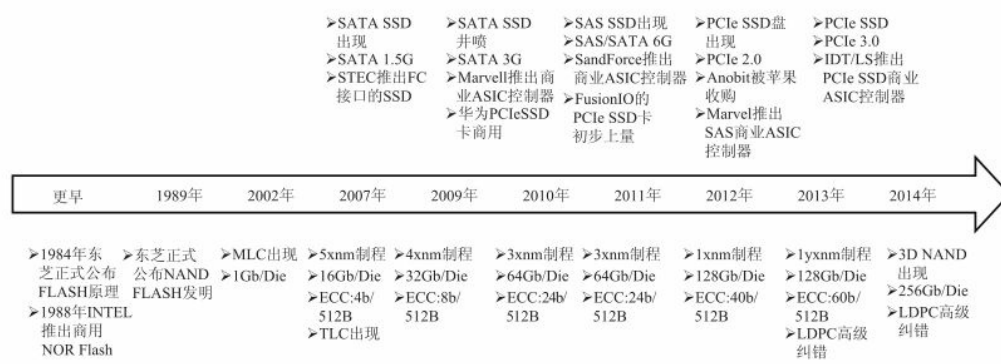


图11-4 NVRAM和PCIe SSD性能规格对比

NVRAM技术

NVRAM非易失性随机访问存储器（Non-Volatile Random Access Memory），是指断电后仍能保持数据的一种RAM。本章节讲述的NVRAM主要针对两种形态，即“DRAM+NANDFlash+备电的NVDIMM”形态和PCIe NVRAM形态，这两种形态以获取与内存相当的访问性能为目标。

NVRAM（NVDIMM、PCIe NVRAM）比SSD具有更低的时延、更大的带宽、更高的IOPS、无写寿命限制等优势，但价格相比SSD更高，容量比较有限，主要用于写Cache应用（见表11-1）。

表11-1 NVRAM和PCIe SSD的性能规格对比

	NVDIMM	PCIe SSD	PCIe NVRAM
存储颗粒	DRAM+SLC Nand Flash	MLC Nand Flash	DRAM+SLC Nand Flash
性能	带宽高：9.4GB/s 时延：<1us级	读带宽：1.1~3.2GB/s 写带宽：0.6~2.8GB/s 4k随机读IOPS：260~720K 4k随机写IOPS：60~240K 写时延：20us 读时延：62us	带宽：3.2GB/s(顺序读写) 4K随机I/O读写 IOPS：630K 时延相对高：<16us
容量	2GB/4GB/8GB	400GB~2.4TB	4GB/8GB
寿命	不限制写	有限写寿命	不限制写
接口	DDR3 RDIMM	PCIe GEN2×8	PCIe GEN2×8
尺寸	标准DIMM条形状，备电 占用1个DIMM空间	全高半长	全高半长
访问方式	内存接口	块设备接口	标准字符设备、块设备； API接口(封装DMA)； 内存接口(MMIO方式)

低延迟、高带宽的网络技术**RDMA/IB**

RDMA（Remote Direct Memory Access）技术全称远程直接数据存取，顾名思义是远程的DMA，通过该技术，可以通过网络把数据直接传入另一台服务器（设备）的某一块内存区域，几乎不需要耗费本地与对端的服务器CPU的处理能力。在实现上，其主要通过将可靠传输协议固化到硬件并且支持零拷贝技术来实现。

目前，有三种高速互联技术（RDMA），分别是老牌技术InfiniBand、基于以太网的RoCE技术以及基于以太网的iWARP技术。

InfiniBand的芯片厂家少，仅有Mellanox、QLogic两家，分别占据80%和20%的市场份额，厂家少的优点是新数据传输速率标准的制定的相应产品的推出远比以太网技术快。InfiniBand技术主要应用于：

- 高端HPC市场；
- 企业级数据中心；

- 金融与证券交易市场；
- Scale-out NAS存储系统及其他高端存储系统；
- 高端数据库系统。

iWARP（Internet Wide-Area RDMA Protocol）是由IETF组织定义的一种能在以太网上使用RDMA技术的网络技术。

RoCE（RDMA over Converged ETH）是由IBTA标准化组织定义的一种在以太网上采用RDMA技术的网络互联技术。

IB、以太和RoCE在时延、带宽成本方面对比情况如下。

- 时延：IB FDR < IB QDR < 40G RoCE < 10G RoCE < iWARP < 40 TCP/IP < 10G TCP/IP。
- 带宽成本：40G RoCE < 10GE < IB FDR < IB QDR。
- 综合对比：40G RoCE最优。

RDMA技术能够出色地降低CPU负载，缩短网络传输时延，并且具有低成本的优势，已经在数据库、HPC、金融、SMB、大数据、虚拟化等场景得到应用。

业界支持RoCE的情况具体如下：

- 开源Linux OFED组件从1.5.1（2010）版本开始支持RoCE，目前已是1.5.4；
- Oracle Exadata/Exalogic一体机通过OFED及IB支持RDMA；
- 微软在SMB2.2时已经支持RoCE，在3.0中更强化了对RDMA的支持。目前，IB、RoCE及iWARP均可在微软平台上使用；
- VMware已开始进行虚拟化环境RDMA支持方面的初步工作；

- EMC的高性能固态存储“Thunder”支持RDMA；
- IBM DB2 pureScale数据库、XIV Gen3存储阵列、TMS固态存储均支持RoCE；
- Emulex会在其Skyhawk（4×10GbE）网络适配器上支持EoCE；
- Mellanox公司Connectx-3网卡产品已经支持RoCE。

各网络协议比较，如表11-2所示。

表11-2 各网络协议比较

协议类型	1/10/40GE	IPoIB	PCIe以太网仿真	iWARP	RoCE	IB	PCIe RDMA
用户态	应用	应用	应用	应用	应用	应用	应用
	Sockets	Sockets	Sockets	Sockets	Sockets	Sockets	Sockets
				verbs	verbs	verbs	verbs
				SDP	SDP	SDP	SDP
				RDMA	RDMA	RDMA	RDMA
内核态	TCP/IP	TCP/IP	TCP/IP	无	无	无	无
		IP over IB	IP over PCIe				
硬件	以太网卡	Infiniband网卡	PCIe网卡	TCP/IP	Infiniband传输层	Infiniband网卡	PCIe驱动
				以太网卡	以太网卡		
	以太网交换机	Infiniband交换机	PCIe交换机	以太网交换机	以太网交换机	Infiniband交换机	PCIe交换机
时延	15~30微秒	15~30微秒	15~30微秒	5微秒	2~3微秒	2微秒	2微秒

重删、压缩技术

目前，数据压缩和重复数据删除是实现数据缩减的两种关键技术。简言之，数据压缩技术通过对数据重新编码来降低冗余度，而重复数据删除技术侧重于删除重复的数据块，从而实现数据容量缩减的目的。

数据压缩和重复数据删除能带来如下好处：节省数据存储空间，降低TCO、节省物理存放空间，提升写入性能，节省网络传输带宽，实现绿色环保（减少电力，空调的使用）。

数据压缩和重复数据删除已经广泛应用于各个领域。

备份领域数据量巨大，要求巨大的数据吞吐率和漫长的备份窗口。传统备份方式的重复数据多，通常可达20：1~30：1，适合使用重删技术。

主存储市场也开始大面积应用重删技术。由于主存储市场关注时延，甚于关注吞吐量，所以多采用后重删技术。指纹查找算法也采用简单的排序查找方法，该方式重删率最高，但耗时巨大，会减少SSD盘的寿命。

虚拟桌面场景VDI在系统盘中有大量的重复数据，可以通过重删来解决空间占用和启动风暴等问题。

即使内存中有大量重复页面存在，但在Linux 2.6.32中引入了KSM（Kernel Samepage Merging）可更加高效地使用内存。

远程复制、广域网加速、字节缓存、源端重删的本质都是先在本地生产数据指纹，然后发送指纹到对端，如果对端已经存在这个指纹，则无需重复发送数据。

下面我们对比分析数据压缩与重复数据删除。

数据压缩和重复数据删除技术都着眼于减少数据量，两者的差别在于：数据压缩技术的使用前提是信息的数据表达存在冗余，而重复数据删除技术的实现依赖数据块的重复出现；数据压缩技术以信息论研究作为发展基础，而重复数据删除技术是一种实践性技术。这两种技术在本质上是相同的，即通过检索冗余数据并采用更短的指针来缩减数据容量。它们的区别关键在于，消除冗余范围不同，发现冗余方法不同，冗余粒度不同，在具体实现方法上也有诸多不同。

数据压缩与重复数据删除两种技术具有不同层面的针对性，并能够结合起来使用，从而实现更高的数据缩减比例。值得一提的是，如果同时应用数据压缩和重复数据删除技术，为了降低对系统的处理需求和提高数据压缩比率，通常需要先应用数据删除技术，然后再使用数据压缩技术进一步降低“结构图”的面积和基本数据块的体积。如果顺序颠倒，会出

现什么样的结果呢？压缩会对数据进行重新编码，从而破坏数据原生的冗余结构，因此再应用重复数据，删除效果会大打折扣，而且消耗时间也更多。而先执行重复数据删除则不同，它首先消除了冗余数据块，然后应用数据压缩对唯一副本数据块进行再次压缩。这样，两种技术的数据缩减作用得到叠加，而且数据压缩的消耗时间大大降低。因此，先去重后压缩可以获得更高的数据压缩率和性能。

11.2.3 一体机产品介绍

业界多个厂家已经纷纷推出多款一体机产品，下面将介绍几款典型产品。

数据库一体机**Exadata**

凭借卓越产品的性能和企业数据库市场的王者地位，Exadata 2012年的销售额约为10亿美元，名列应用一体机市场首位，当前最新产品为第4代Exadata X3。

甲骨文公司在2008年率先推出了业界首款软硬件集成设计的系统——Oracle Exadata数据库云服务器，采用Oracle数据库软件和惠普的硬件，包括服务器、存储、网络设备。2009年甲骨文收购Sun之后推出第二代Exadata V2，全面采用了来自原Sun公司的服务器和存储等硬件架构，并在一体机中配备了当时非常先进的40Gbps速率的InfiniBand和PCIe闪存卡。2010年甲骨文推出的Exadata X2有两个型号，即X2-2（数据库采用2路服务器硬件）和X2-8（数据库采用8路服务器硬件）。另外，Exadata X2具有硬件数据库加密功能，其每个节点的内存可达2TB。

2012年10月甲骨文发布了第四代Exadata X3，并称之为内存数据库机（Database In-Memory Machine），Exadata X3大幅增加了内存和闪存容量，并推出Smart Flash Cache技术，以提升系统性能。与Exadata X2类似，Exadata X3有两个型号，即X3-2和X3-8。

Exadata X3-2一体机的特点如下。

- 容量更大，每个X3-2机架可配8个DB节点，CPU达128核，内存达2TB，40×10Gb带宽；可配置14个存储节点，168个CPU核心进行SQL处理（通过Oracle独有的存储卸载技术）；可配置56个PCI闪存卡，组成共计22.4TB闪存容量；可配置168块600GB的15K RPM高

性能磁盘，或者168个3TB的7.2K RPM大容量磁盘。

- 性能更高，智能Flash Write Cache可将写性能提升20倍，数据吞吐量（SQL查询）达100GB/s。
- 更节能，每个X3-2机架最大可节省3KW电能消耗。

Exadata X3关键软件技术：

- ASM on Exadata，单一表空间，减少90%的存储管理需求；
- SmartScan，可提升10x~100x查询速度；
- EHCC混合型列压缩，信息密度提升5x~50x，并提升查询速度；
- 扫描BW + Storage Index，减少90%以上索引；
- Oracle 11G数据库内存管理、SQL管理、DOP、并发排队等新技术，减少80%以上调优工作；
- 云特性，确保向数据库云方向发展，符合未来技术发展趋势。

基础设施一体机**Vblock**

Vblock是VMware、EMC和Cisco三家厂商合作推出的基础设施一体机，其中服务器和TOR交换机为CISCO产品，存储为EMC产品，虚拟化软件平台为VMware产品，Vblock由EMC、Cisco、VMware与Intel合资的VCE公司负责销售与维护。

第一代的Vblock于2010年中推出，最初分为Vblock 0、Vblock 1与Vblock 2三个层级，分别涵盖低、中、高阶应用，存储设备分别采用EMC的Celerra、Clariion及Symmetrix VMAX等型号。Vblock在2011年中做了改款，推出Vblock 300系列取代了Vblock 0、1与1U，并将Vblock 2更名为Vblock 700，于是Vblock家族便简化为Vblock 300与Vblock 700两个系列。新的Vblock 300系列改用EMC新推出的VNX系列储存设备，VNX是一种能同时支持NAS、FC与iSCSI SAN等不同类型存取功能的多

协议储存设备，因此以往的Vblock 1与1U的区别也没有必要存在。

2012年VCE基于新一代Intel CPU和EMC存储产品推出Vblock 320替代Vblock300，Vblock 720替代Vblock 700，同时推出针对SMB客户的Vblock 100和Vblock 200（见表11-3）。

表11-3 基础设施一体机对比

	Vblock 100	Vblock 200	Vblock 320	Vblock 720
应用场景	中小企业/远程/分支机构(数据中心/VDI)	中型企业(中型数据中心/核心IT基础设施/VDI/微软应用/数据库)	云服务提供商/企业(企业关键应用如微软应用/Oracle/SAP/SAPHANA)	云服务提供商/企业(企业关键应用如微软应用/Oracle/SAP/SAPHANA)
服务器数量	BX(3-4)/DX(3-8)		最大128	最大384
CISCO服务器	机架式服务器 C200 M3		半宽/全宽刀片	半宽/全宽刀片
EMC存储	VNX3150/ VNX3300	VNX5100	VNX5300/5500/ 5700/7500	VMAX 10K/20K/40K
系统规模	1(24U或42U机柜)	1(42U机柜)	1-8(42U机柜)	1-28(42U机柜)
计算网络(CISCO)	2×24口3750-X	2×Nexus 5548UP(10GE)	Nexus 5548UP/5596UP(10GE)	Nexus 5548UP/ 5596UP(10GE)
存储网络(CISCO)			MDS 9148(FC)	MDS 9148/9513(FC)

除了搭配VMware的vCenter来管理系统外，EMC专门为Vblock开发了Ionix Unified Infrastructure Manager管理工具统一管理Vblock的硬件资源分配，并与VMware vCenter、vCloud Director整合，提供对基础设施的资源分派、监控与管理功能。

基础设施一体机FusionCube

华为FusionCube是一款将计算、存储、网络进行垂直整合的融合架构硬件平台，其在12U的机框内已经预集成了40GE/100GE的以太网和InfiniBand，背板总带宽可达15.6Tbps,为整体融合架构提供了足够的网

络带宽和性能保证。1.5U高的刀片设计保证了单刀片的内存容量两倍于业界同类产品；半宽刀片（2路四核到8核）的设计大大提升了计算密度，亦两倍于业界同类产品。FusionCube采用存储刀片，配合DSware分布式存储引擎，可替代传统的大集中SAN架构，同时解决计算和存储的灵活配比的问题。

相对Vblock，华为FusionCube有两个融合。

（1）计算和存储的融合

FusionCube方案取消了存储的机头，直接用服务器控制存储，这是计算和存储的融合，数据存取速度更快。

（2）网络的融合

Fusioncube交换机融合在背板上，通过背板的InfiniBand总线把计算和存储进行了网状直联，速度高达56Gbps。

FusionCube还能做到统一管理。一般来说，IT管理运维的成本占到整体成本的65%，FusionCube能对软件、硬件、虚拟化进行统一管理，大大减少运维成本。华为的FusionManager云管理软件包含了功能强大的资源管理，同时提供了自动发现、安装、部署等功能，为融合架构提供了统一的管理平台。除此之外，华为云管理软件FusionManager还集成了众多的自动化运维功能，如一键式部署、基于策略的管理等，这大大提高了运维效率，降低了成本。

基础设施一体机Nutanix

初创厂商Nutanix提供的一体机产品为融合架构一体机，将服务器、存储和网络集成到一台服务器。每个节点都是一台x86服务器，每个节点采用相同的构造，支持多节点叠加构建大型的数据中心；该部分采用了Nutanix分布式存储和分层存储技术，热数据存储在高性能的PCIe闪存卡上，常用数据存在SSD上，而不常用的数据则会被存在磁盘上。

Nutanix一体机产品具有简单、软件定义以及高可扩展三个特点。

（1）Nutanix一体机简化一体机的结构。计算与存储资源融合于单一的集成平台，使得应用和虚拟化团队能够迅速、简便地部署新的虚拟机，

而无需配置后端存储系统。这也是Nutanix融合理念的一部分，由于融合，所以简单，由于简单，所以降低成本。

（2）其次是软件定义，Nutanix的软件架构专门为虚拟化而设计，并支持各种流行的技术，包括实时VM迁移、高可用性（HA）、分布式资源调度（DRS）以及容错技术等。同时，Nutanix软件架构还不限定管理程序，可支持VMware vSphere和KVM，以及各种行业标准的管理系统。整个的Nutanix虚拟计算平台在其软件架构的支持下可以实现一种全面集成的系统，可应对任意规模的虚拟工作负载。

（3）高可扩展。Nutanix宣称架构无需停机就可以无缝地添加其他Nutanix节点，并且由于都采用同样的节点，所以扩展操作十分简便。

11.3 一体机市场、技术趋势

11.3.1 一体机市场趋势

一体机是模块化数据中心的建设基石

基础设施的业务价值取决于其支撑的IT业务需求。随着新技术如虚拟化和云计算的运用，各种业务流逐渐IT化，数据中心的数据量因此快速增长。数据中心的发展聚焦于：能够快速部署业务，数据中心内各厂家设备能够统一维护，数据中心基础设施能扩容，数据中心能维持高水平的资源利用率，实现更低的TCO。

传统数据中心将来自不同供应商的、类型迥异的组件进行定制设计，在现场组合成一个独特的大型基础设施系统。由于这种方式工作量大，周期长，基础设施维护复杂，调整困难，因此无法满足IT业务对于数据中心的需求。近年来，客户对于模块化数据中心的呼声越来越高。

一体机产品出现后，由于其部署周期短，维护统一，易于扩展，业务预集成、调测等特点，正适合作为数据中心的模块化建设的基石。业界普遍预测，一体机年增长率将达到40%，远高于数据中心各类分散产品（如服务器、存储、网络设备）的增长率。到2016年，一体机市场空间将达到约180亿美元，约占整个数据中心市场的15%。

一体机需针对各类客户提供差异化产品形态

（1）中小企业（简称为SMB）

这类企业的IT设施规模不大，但IT应用种类丰富，IT维护人员少，对成本和设备的性价比很敏感。因此，SMB需要小规模、架构简单、性价比高的一体机产品。

（2）大企业

这类企业的IT设施规模较大，IT应用规模和种类较多，对于IT设施的可靠性、扩展性、维护性要求较多，拥有专业的、经验较为丰富的IT维护人员。大企业往往需要较大规模、架构容易扩展、可靠性高的一体机产品。

（3）服务提供商

服务提供商主要指向企业或最终客户提供电信业务、互联网业务、数据中心外包等业务的提供商，这类企业的IT设施规模大，提供专业的服务和应用，需要帮助其快速推出业务，应对业务快速增长，为最终客户提供高性能体验的IT基础设施。服务提供商本身拥有大量专业的、经验丰富的各种设施IT维护人员。

服务提供商往往需要大规模、分布式架构、容易扩展、可靠性高、能快速推出业务的一体机产品，例如由高密度服务器、分布式存储、大流量交换机组成的大型一体机，并且可以根据应用需求定制专门的一体机产品。

（4）超大规模业务提供商

超大规模业务提供商主要指一些全球性的业务提供商，例如搜索巨头Google，电子商务兼云计算服务提供商Amazon，这些全球性IT业务巨头拥有大量IT研发工程师和维护工程师，可自己开发业务系统，因此往往需要定制硬件，通常不会采用一体机。

未来一体机的市场增长主要来源于有持续需求的大企业、数量众多的SMB和需要超大规模一体机的服务提供商。

一体机需满足各种类型企业应用

构筑于数据中心之上的企业IT信息系统需支持各种类型IT应用，例如

VDI应用。各类应用对于性能需求不同，对于计算、网络、存储资源要求和可靠性要求也不同。例如由于Web Server的网络流量大，连接数多，因此对网络资源消耗大，而对计算能力要求不高；由于应用服务器需要处理各种业务逻辑，因此对计算能力要求高，对网络资源消耗相对较少。

一体机需具备以下特点：

- 灵活分配各类资源，满足各类应用的不同资源需求；
- 具有业务SLA保障机制，保证各类业务的服务质量；
- 快速部署业务的能力，例如业务部署模板；
- 灵活扩容业务的能力；
- 完善的可靠性机制，以最优性价比方案提供各种应用对于RTO/RPO的需求。

11.3.2 一体机技术趋势

新硬件介质

随着时间演进，新介质存储容量持续高速增长；而随着技术及工艺的演进，新介质性能向更好的方向发展。

未来对DDR RAM形成替代的技术包括STT MRAM、PCM和FeRAM，未来对Nand Flash形成替代的技术包括PCM和ReRAM。

ReRAM结构简单，易于进行3D堆叠，可以实现大容量产品。其读写速度、Endurance优于NAND，但低于STT-MRAM，主要定位为取代NAND Flash。当前阶段还没有商业化的ReRAM产品，预计最早的商业产品会在2015年出现。ReRAM容量小、成本高，主要在NV Cache、集成内存、价格不敏感等应用场景下使用。在2017年以后，ReRAM产品有望达到TB级的容量，同时价格低于NAND，从而成为主要大容量存储介质。

PCM当前的主要产品是Nor Flash接口，最新的45nm PCM性能较65nm Nor Flash提升较多，成本较SRAM有较大优势（每GB4.5美元），价格约为SLC的10倍左右（NAND Flash SLC成本为每GB4.5美元）。目前，PCM在智能电表等对写速度要求不高的场景中可以取代串行SRAM。

STT-MRAM具有高数据刷新率、无限次存写、高读写速度以及低能耗和非易失性等特点，有望取代DRAM，预计STT-MRAM上市初期目标可接近DRAM的成本，比NAND的成本高出10倍（见表11-4、表11-5）。

表11-4 数据易失类型与非易失类型

类 型	说 明	
易失类型 (掉电丢失)	大规模量产	SRAM、DRAM、PSRAM
	起步阶段(特定场景)	浮体内存
非易失类型 (掉电不丢失)	大规模量产	NOR、NAND、EPROM、EEPROM、Mask ROM
	小规模量产	MRAM、FRAM
	起步阶段(特定场景)	PRAM、MRAM、PRAM、赛道内存

表11-5 存储介质时延、擦写次数比较

分 类	读 写 时 延	P/E Cycle(编程/擦除周期)	备 注
SRAM	5纳秒/5纳秒	10 ¹⁵	可按字节设定地址
STT-MRAM	20纳秒/20纳秒	10 ¹⁵	
DDR3	30纳秒/15纳秒	10 ¹⁵	
PCM	50纳秒/120纳秒	10 ⁸	

(续表)			
分 类	读 写 时 延	P/E Cycle(编程/擦除周期)	备 注
ReRAM	50微秒/50纳秒	10^8	可按地址块设定地址
PCM-S	50纳秒/120纳秒	10^6	
SLC	25微秒/200微秒	10^5	
Emlc	50微秒/1200微秒	10^4	
MLC	50微秒/800微秒	10^3	
SAS 15k	3毫秒/3毫秒	10^8	
SAS 10k	3毫秒/5毫秒	10^8	

融合和简化

IT系统的发展演变从最初的大一统分化成服务器、存储、网络等。如今，融合的趋势将越来越明显。

融合的概念最早是指一个集服务器、网络、存储以及管理软件于一体的整体解决方案，这一理念迅速被业内接受，目前几乎主流的厂商均推出具有各自技术特点的融合基础架构（或者说“一体机”）产品，并且这一市场也开始悄然兴起。据IDC在2013年的十大预测中提到，在未来几年，企业数据中心服务器65%都会采用融合基础设施的设备。可以说，这一市场前景广阔。

一体机中的融合技术有几个方向。

（1）计算/存储融合（代表厂商：华为FusionCube）

华为FusionCube采用独有的计算和存储融合技术，将标准服务器上本地存储组成虚拟的SAN存储资源，无需外置SAN设备就可以提供计算和存储能力。

计算存储融合带来了更高性能。在传统的IT架构中，服务器与存储分离导致计算节点只能通过机头以串行的方式访问数据，效率较低。

FusionCube一体机通过在刀片服务器中部署分布式存储引擎，将计算节点串行访问数据的方式改为并行访问，由此能够同时从多个节点的本地存储中获取数据，减少访问时延。

计算存储融合可实现管理简单与一体化运维。以往，企业在进行IT基础架构设计时，需要考虑服务器选型、软硬件兼容性验证等工作，会耗费

大量的时间和精力。而在FusionCube一体机中，企业看到的不再是单独的服务器、存储设备、网络，而是一个软硬件资源深度整合的融合系统，由此减少方案设计的工作量，降低方案设计的复杂度。到了建设环节，FusionCube一体机已经实现软硬件的预集成、预安装和预验证，可一次性完成安装部署，改变了以往不同厂商多次现场安装设备的方式，降低了人力资源的成本，现场安装时间也由原来的7天缩短为2小时，实现“家电式”安装。在预集成方面，FusionCube一体机直接把产品的操作经验固化成模板，然后内置于IT系统，允许系统自动处理各种基础而耗时的的工作，从而将IT应用部署的时间缩短为6小时，运维工作量减少60%。正是由于FusionCube一体机将软硬件系统充分地融合在一起，IT运维人员不再需要特别关注某一个软件或硬件，而更聚焦于IT系统的整体性能和用户体验。

（2）计算和网络融合（代表厂商及产品：CISCO，UCS）

CISCO是老牌的网络厂商，在CISCO推出的UCS产品中充分发挥了其网络优势。

➤ 可靠性

UCS的架构与普通刀片不一样。刀片机箱内无交换设备，刀片机箱的网络连到Cisco UCS 6120XP端口互联阵列设备上，6120XP工作在end-host模式。在配置刀片和6120XP连接时，用户可以指定某块网卡连到特定的6120XP。这条链路上任何一个部件出现故障（6120XP本身、机箱到6120XP的连线、6120XP到上行交换机的连线或者是上行交换机本身），刀片服务器上的网卡会自动切换到另一条链路，而且切换过程不丢包，保证业务的连续性。

（3）简化运维管理

UCS采用了Service Profile和无状态计算概念，所有刀片服务器在没有配置前均可视为裸机，因此刀片服务器的物理位置或者插入哪个机箱已经不再重要，这样就为跨机箱的服务器冗余做好物理上的准备。

当用户设置了Server Pool，并将某个配置文件关联到此Server Pool时，该配置文件会自动寻找第一个可用的服务器资源，并与它做关联。当该服务器损坏时，配置文件会继续寻找第二个可用服务器资源并自动与之关联。由于配置文件中规定了服务器的物理参数，如MAC地址、WWN

地址、VLAN和vSAN的连接、boot order等各项参数，新的服务器（也称备机）就会具有和老服务器一样的物理参数，由此，无需在网络和SAN设备上做任何重新设置，备机就具有老服务器的所有特性，如果采用boot from SAN的模式，OS和用户应用就可自动重启，完全无需人工干预。

（4）应用和基础设施的融合（代表厂家和产品：HP，CloudSystem）

类似机架或者刀片等通用服务器在一定程度上并不能帮助用户实现效率最大化，效率必须与应用挂钩。Windows、ERP、电子邮件或者数据库应用对于性能的需求是不一样的，有的应用可能要求内存较大，有的则需要存储空间较大，而有些则可能对带宽有较高要求，通用服务器并不能完全满足这些细致的需求。尽管现在服务器厂商都在一定程度上提供个性化选择，但这并没有做到极致。

惠普希望在这方面的创新能够帮助用户最大程度地优化IT资源，例如大规模采用ARM架构的服务器，在同一块主板上集成6个服务器的高密度服务器等（编者注：这并不代表惠普未来会推出具体的产品），但惠普更加强调的是与应用结合，即为用户提供最佳实践。

将IT基础设施与应用相结合必然要涉及第三方软件服务提供商

（ISV），在通用软件方面，惠普与广泛的ISV有着深入的合作。在这方面，惠普提出帮助用户快速建立应用系统架构的口号，通过了解用户的实际应用需求，结合惠普的经验和整个业界的最佳实践，为用户提供模板信息，例如构建一个具有一定规模的Web服务需要的CPU数量、内存大小、存储空间以及带宽大小等建议信息。在应用层面，可以通过应用直接在资源池分配资源部署应用，无需用户再去具体的物理机上安装应用。

（5）开放/标准化

在一体机被市场接受的过程中，其是否会被供应商锁定的质疑也是一直存在。业界和厂家在试图朝开放接口、开放标准的方向来解决这个质疑。

支持OpenStack一体机的厂家代表产品为华为FusionCube。

华为FusionCube采用OpenStack的架构，把自己的存储、服务器、虚拟

化产品做开放式接入。从客户角度来看，不需要像以前那样全套购买华为的产品，因为通过OpenStack，第三方软件可以通过调用OpenStack API来提供服务。OpenStack可通过各个模块调用华为或第三方提供的存储、网络、虚拟化等能力。比如，用户可以同时使用如VMware的Hypervisor和华为的Hypervisor。这样，在OpenStack的架构里，华为只是其中一部分的能力提供方，客户不会被一家所绑定。

开源和开放的API不仅让用户免于Lock-in，同时能减少运维等维护复杂度，降低成本。例如在运维方面，通过开放的API，用户可以选择第三方的工具或开源软件，让企业的基础设施管理不会被绑定。同时，企业能够灵活享受云化后带来的好处，降低云化之后带来的复杂性。

结语

云计算的概念已经炒作了很多年，各种云计算的相关概念风生水起，世界各大IT公司，各种创业公司都投入云计算中，但是云计算目前的实施效果并不如意，对于云计算的准确样子仍然没有一个明晰的定论。然而，不管有什么困难，有一点始终像启明灯一样地照耀着我们，那就是云计算的终极愿景，希望人们像用水用电一样随时随地地使用信息服务。只要坚持这个愿景，以这个愿景来衡量，一切与云计算相关的概念和技术是否符合历史的潮流都会一目了然。

本书的范围仅是聚焦于云计算的IaaS层，聚焦于提供基础设施的厂家如何构造一个满足云计算需要的共享资源池，这个共享资源池能够给上层的平台层、应用层提供随要随给的资源，以实现云计算的终极目标。那么，要满足这样的目标，有几大技术是基础的，也是必须的。

这些技术具体如下。

- 计算虚拟化：IT服务的核心就是提供计算服务，如何最好地将各种小的物理计算资源构成大的计算池，随之又按照需要分配给申请者，并进行动态的资源管理。
- 存储虚拟化：IT服务的另外一个核心就是存储，所有的信息必须能够保存下来才有意义。保存分为暂时的保存和永久的保存。存储方面又有几大必须解决的难题：
 - 如何提高I/O能力，保证尽可能大的读写带宽；
 - 如何做到不同存储厂家异构资源的融合；
 - 如何针对不同的应用智能满足客户的需求，存储资源是有限的，存储的I/O无论如何是有瓶颈的，恰到好处的服务就是对存储资源的最好利用。
- 网络虚拟化：当所有的企业和个人都随时随地地访问云服务时，网络的复杂性一定是以指数增长的。那么，这个虚拟网络如何建立，需要数学模型的计算。对于安全性，亦是虚拟化网络的重中之重。

➤ 虚拟化安全：随时随地的访问，需要无处不在的安全防护。接入访问需要安全，数据在云网络中流动需要安全，云数据在CPU处理时需要安全，云数据在存储池中保存时也需要安全。

➤ 分布式集群：随时随地的访问意味着提供基础服务的硬件设施一定是分布式集群组成的系统，否则不可能给所有访问的用户提供服务。这个集群需要多大，采用多少层级，如何进行区域性的调动，都是需要解决的难题。

➤ 异构资源融合或标准化：目前有很多的IT硬件厂家，他们所研发的服务器、存储设备、网络设备成千上万，异构资源必须能够融合，否则云计算的共享资源池无从谈起。融合有两条路，第一条，每个厂家都遵循一个接口规范来设计和制造服务器、存储设备和网络设备；第二条，设计一个计算、存储、网络融合的单一设备，这个单一设备有一致的接口，可以像搭标准积木一样部署出超大的云计算资源系统。

展望未来，在云计算基础设施领域有几大技术趋势，这几个技术的深度实现将有助于云计算真正走向千家万户。

➤ 内存计算：在未来的世界，实时计算、实时决策才是王道，而要达到如此的效果，必然会走向内存计算。所有的数据和计算全部保存在内存中，通过内存集群和高速网络保证高性能；数据有多个异地副本，可解决可靠性问题；传统的基于磁盘或者磁带的存储，在系统正常运行时不再参与到主业务流程中，更多的是作为一个备份的介质，在内存突然全部故障的情况下，将内存数据全部恢复出来。

➤ 超大集群技术：不管是单个大型跨国企业，还是提供云服务的云服务商，在现代信息爆炸的时候，需要的系统规模都会越来越大。那么，如此巨大的系统如何高效安全地建立连接，如何高效快速地进行资源的发放和回收，如何保证系统整体的最优，如何给出一个有效的建模和仿真，这些都是很困难的问题。

➤ 物联网技术：即使云计算平台作为一个计算中心解决了数据计算和决策的过程，但是如果没有对各种终端设备的有效接入和连接，我们仍然无法做出对现实世界实时的反应，因此物联网技术也

是云计算走向深入发展后的必然需求，例如汽车上的各种移动传感器、个人身上携带的各种健康和移动传感器、家用电器的统一健康状态监控传感器、市政基础设施的健康传感器（路灯状态、红绿灯状态、井水盖状态、下水道状态等）、智能楼宇的各种传感器（水、电、煤气等的使用和健康状态），等等。当各种各样的接入终端设备的数据源源不断地传送到数据中心后，基于内存大集群的云数据中心就会非常快速地计算出系统的故障所在，给出故障纠正措施，并通过云网络反馈到智能终端，迅速做出调整，消灭故障或者避免更大的事故发生。

缩略语

缩 略 语	全 称	说 明
AAA	Authentication、Authorization、Accounting	验证、授权和记账
ACL	Access Control List	接入控制列表
ACM SIGCOMM	ACM: Association for Computing Machine, 美国计算机协会 SIGCOMM: Special Interest Group on Data Communication, 数据通信专业组	SIGCOMM是ACM组织在通信网络领域的旗舰型会议, 也是目前国际通信网络领域的顶尖会议, 由ACM SIGCOMM组织举办
AD	Active Directory	活动目录
Aero	Authentic(真实)、Energetic(动感)、Reflective(反射)及Open(开阔)	Windows Aero是从Windows Vista开始使用的新型用户界面, 透明玻璃感让用户一眼贯穿
API	Application Programming Interface	应用程序接口
ARM	Advanced RISC Machine	一种处理器架构
ARP	Address Resolution Protocol	地址解析协议
ASM	Any-source multicast	任意源组播
ATOM	Intel® Atom™	英特尔® 凌动™处理器, 是Intel的一个超低电压处理器系列
BIOS	Basic Input/Output System	基础输入输出系统
BOSS	Business Operating Support System	业务运营支撑系统
BSS	Business Support System	业务支撑系统
CBT	Changed Block Tracking	改变数据块跟踪
CEP	Complex Event Processing	复杂事件处理
CHAP	Challenge Handshake Authentication Protocol	询问握手认证协议
CIM	Common Information Model	通用信息模型
CLOS	英文姓氏	一种无阻塞交换架构的名称, 源于发明者Charles Clos博士
CORBA	Common Object Request Broker Architecture	通用对象请求代理架构是软件构建的一个标准
CQL	Continuous Query Language	连续查询语言
CRM	Customer relationship management	客户关系管理

(续表)

缩 略 语	全 称	说 明
DaaS	Desktop as a Service	桌面即服务
DAS	Direct-Attached Storage	直接附加存储
DB	DataBase	数据库
DC	Data Center	数据中心
DDoS, DOS	Distributed Denial of Service	分布式拒绝服务攻击
DevOps	英文Development和Operations的组合	DevOps是一组过程、方法与系统的统称,用于促进开发(应用程序/软件工程)、技术运营和质量保障(QA)部门之间的沟通、协作与整合;它的出现是由于软件行业日益清晰地认识到,为了按时交付软件产品和服务,开发和运营工作必须紧密合作
DHCP	Dynamic Host Configuration Protocol	动态主机配置协议
DHT	distributed Hash table	分布式哈希表
DPM	Distributed Power Management	分布式电源管理
DRS	Distributed Resource Scheduler	动态资源调度
DSP	Digital Signal Process	数字信号处理器
DX	DirectX(Direct eXtension)	微软公司建立的一系列专为多媒体以及游戏开发的应用程序编程接口
EBS	Elastic Block Store	弹性块存储
EC2	Elastic Compute Cloud	弹性计算云
ECC	Error Checking & Correction	错误检查与纠正
ECMP	Equal-Cost Multi-Path routing	等价多路径路由
EDC	Enterprise Data Center	企业数据中心
EOI	End of Interrupt	中断结束
EPT	Extended Page Table	页表扩充技术
ERP	Enterprise Resource Planning	企业资源计划
ESP	Event Stream Processing	事件流处理
ETL	Extract-Transform-Load	用来描述将数据从来源端经过萃取(extract)、转置(transform)、加载(load)至目的端的过程
EVS	Elastic Virtual Switch	虚拟交换机
FC	Fibre Channel	光纤通道
FCAPS	Fault, Configuration, Accounting, Performance, Security	表示网络管理的五种基本功能的缩写:故障、配置、计费、性能、安全管理
FW	FireWall	防火墙
GDI	Graphics Device Interface	图形设备接口
GDT	Global Descriptor Table	全局描述符表

(续表)

缩略语	全 称	说 明
GFS	Google File System	Google文件系统
GPA	Guest OS Physical Address	客户操作系统的物理地址
GPS	Global Positioning System	全球定位系统
GPU	Graphics Processing Unit	图形处理器
GUI	Graphical User Interface	图形用户界面
GVA	Guest OS Virtual Address	客户操作系统的虚拟地址
HA/FT	High Availability/Fault Tolerance	高可用性/容错
HDD	Hard Disk Drive	硬盘
HDFS	Hadoop Distributed File System	Hadoop分布式文件系统
Hop	Hadoop online Prototype	Hadoop在线原型
HPA	Host OS Physical Address	宿主操作系统的物理地址
HPC	High Performance Computing	高性能计算
HRM	Human Resource Management	人力资源管理
HTML	HyperText Markup Language	超文本标记语言
HTTP/HTTPS	HyperText Transfer Protocol Secure	超文本传输(安全)协议
I2RS	Interface to Routing System	路由系统接口
IaaS	Infrastructure as a Service	基础设施即服务
ICA	Independent Computing Architecture	独立计算架构, 思杰公司开发的远程桌面协议
ICP	Internet Content Provider	互联网内容提供商
IDS	Intrusion Detection Systems	入侵检测系统
IDT	Interrupt Descriptor Table	中断描述符表
IETF	Internet Engineering Task Force	互联网工程任务组
IF	Interface, IF1 IF2 IF3...	接口缩写代号
iNIC	Intelligent Network Interface Card	智能网卡
I/O	Input and Output	输入输出
IOMMU	Input/Output Memory Management Unit	输入输出内存管理单元
IOPS	Input/Output Operations Per Second	每秒读写(I/O)操作的次数
IPAM	IP Address Management	IP地址管理
IPMI	Intelligent Platform Management Interface	智能平台管理接口
IRF	Intelligent Redundant Framework	智能冗余框架
IS-IS	Intermediate System-to-Intermediate System	中间系统到中间系统
ISP	Internet Service Provider	互联网服务提供商

(续表)

缩 略 语	全 称	说 明
ISV	Independent Software Vendor	独立软件开发商
ITIL	Information Technology Infrastructure Library	信息技术基础设施库
ITSM	IT Service Management	IT服务管理
KPI	Key Performance Indicators	关键绩效指标
KSM	Kernel Samepage Merging	内核同页合并
LAMP	Linux, Apache, MySQL, PHP	LAMP是指一组通常一起使用来运行动态网站或者服务器的自由软件名称首字母缩写：其中L指Linux，为操作系统，A指Apache，为网页服务器，M指MariaDB或MySQL，为数据库管理系统(或者数据库服务器)，P指PHP、Perl或Python，为脚本语言；PHP用于编辑网页，网页运行在Apache软件上，网页的数据来自MySQL，Apache和MySQL安装在Linux操作系统上
LB	Load Balance	负载均衡
LBS	Load Balance Server	负载均衡服务器
LDAP	Lightweight Directory Access Protocol	轻型目录访问协议
LUN	Logical Unit Number	逻辑单元号
MBPS	Megabyte Per Second	兆字节每秒
MME	Multi Media Extension	多媒体扩展
MMU	Memory Management Unit	内存管理单元
MPP	Massively Parallel Processing	大规模并行处理
NAS	Network Attached Storage	网络附属存储
NFS	Network File System	网络文件系统
NFV	Network Function Virtualization	网络功能虚拟化
NOS	Network Operation System	网络操作系统
NUMA	Non-uniform Memory Architecture	非一致性内存架构
NVDIMM	Non-Volatile Dual In-line Memory Module	NVDIMM是在一种集成了DRAM和非易失性内存芯片的内存条规格，能够在完全断电的时候依然保存完整的内存数据
NVGRE	Network Virtualization Using Generic Routing Encapsulation	采用通用路由封装的网络虚拟化
NVRAM	Non-Volatile Random Access Memory	非易失性随机访问存储器
OA	Office Automation	办公自动化

(续表)

缩略语	全 称	说 明
OAM	Operations, Administration, and Maintenance	运营、管理、维护
OBS	Object-Based Storage	对象存储
OLAP	On-Line Analytical Processing	在线分析处理
OLTP	On-Line Transaction Processing	联机事务处理
OM	Operation Management	运营管理
OMS	Operation Management System	运营管理系统
ONF	Open Networking Foundation	开放网络基金会
OpenGL	Open Graphics Library	开放图形库定义跨程序语言、跨平台的应用程序接口(API)的规范，用于生成二维、三维图像
OS	Operation System	操作系统
OSS	Operations Support System	运营支撑系统
OVF	Open Virtualization Format	开放虚拟化格式
OVSDB	Open vSwitch Database Management Protocol	开放虚拟交换机数据库管理协议
PaaS	Platform as a Service	平台即服务
PCB	Printed Circuit Board	印刷电路板
PCI	Peripheral Component Interconnect	外设互联标准或个人电脑接口
PCR	Platform Configuration Register	平台配置寄存器
PDA	Personal Digital Assistant	个人数码助理
PDM	Product Data Management	产品数据管理
PF	Physical Function	物理功能
PIN	Personal Identification Number	个人鉴别码
PM	Physical Machine	物理主机
POSIX	Portable Operating System Interface	可移植操作系统接口
PUE	Power Usage Effectiveness	能源使用效率
PXE	Preboot eXecution Environment	预启动执行环境
QEMU	Quick EMUlator	一款仿真处理器的自由软件
QoE	Quality of Experience	用户体验质量
QoS	Quality of Service	服务质量
RAID	Redundant Array of Independent Disks	独立硬盘冗余阵列
RAS	Reliability, Availability and Serviceability	可靠性、可用性、可维护性
RBAC	Role Based Access Control	角色的访问控制

(续表)

缩略语	全 称	说 明
RDB	Relational Database	关系型数据库
RDMA	Remote Direct Memory Access	远程直接数据存取
RDP	Remote Desktop Protocol	远程桌面协议
RESTFUL	REST, Representational State Transfer	含状态传输的 Web 服务(也称为 RESTful Web API)是一个使用HTTP并遵循REST原则的Web 服务
RoCE	RDMA over Converged Ethernet	—
RoCE	RDMA Over Converged ETH	—
RPO	Recovery Point Objective	恢复点目标
RTP	Real-time Transport Protocol	实时传输协议
S4	Simple Scalable Streaming System	—
SaaS	Software as a Service	软件即服务
SAN	Storage Area Network	存储区域网络
SAS	Serial Attached SCSI	一个用于数据存储的点对点串行协议
SATA	Serial ATA	一种计算机总线接口, 一般用于硬盘
SBC	Server-Based Computing	应用虚拟化
SDN	Software-Defined Networking	软件定义网络
SDS	Software-Defined Storage	软件定义存储
SGC	Security Gateway Controller	安全网关控制器
SGSN	Serving GPRS Support Node	GRPS服务节点
SLA	Service-level Agreement	服务等级协议
SMB	Small and Medium-sized Business	中小企业
SMI-S	Storage Management Initiative Specification	存储管理初始化配置
SMP	Symmetric Multiprocessing	对称多处理
SNAT	Source Network Address Translation	源地址转换
SNMP	Simple Network Management Protocol	简单网络管理协议
SOA	Service-oriented Architecture	面向服务的架构
SOAP	Simple Object Access Protocol	简单对象访问协议
SPB	Shortest Path Bridging	最短路径桥接
SQL	Structured Query Language	结构化查询语言
SR-IOV	The Single Root I/O Virtualization	一种基于硬件的虚拟化解决方案
SSD	Solid State Disk、Solid State Drive	固态硬盘
SSH	Secure Shell	安全命令解释器

(续表)

缩略语	全 称	说 明
STT	Stateless Transport Tunnelling Protocol	无状态传输协议
TC	Thin Client	瘦客户端
TCM	Thin Client Management	瘦客户端管理
TCO	Total Cost of Ownership	总拥有成本
TCP	Transmission Control Protocol	传输控制协议
TLV	Type-Length-Value	类型 - 长度 - 值
TPM	Trusted Platform Module	可信平台模块
TRILL	Transparent Interconnection of Lots of Links	多链接透明互联
TXT	Trusted Execute Technology	可信执行技术
UC	Unified Communications	统一通信
UDP	User Datagram Protocol	用户数据报协议
UEFI	Unified Extensible Firmware Interface	统一的可扩展固件接口
UVP	Universal Virtualization Platform	华为公司的虚拟化平台
vCPU	Virtual CPU	虚拟CPU
VDI	Virtual Desktop Infrastructure	虚拟桌面基础设施(桌面云)
VF	Virtual Function	虚拟功能
VGA	Video Graphics Array	视频图形阵列
VIF	Virtual Interface	虚拟接口
VIP	Virtual IP	虚拟IP
VM	Virtual Machine	虚拟机
VMCS	Virtual Machine Control Structure	虚拟机控制架构
VMDq	Virtual Machine Device Queue	虚拟设备队列
VMM	Virtual Machine Manager	虚拟机管理器
VNC	Virtual Network Computing	虚拟网络计算
VNI	VxLAN Network Identifier	VxLAN网络ID
VoIP	Voice over Internet Protocol	IP电话
vPC	Virtual Port-Channel	虚拟端口通道
VPC	Virtual Private Cloud	虚拟私有云
VPLS	Virtual Private LAN Service	虚拟专用局域网业务
VPN	Virtual Private Network	虚拟专用网
VR	Virtual Reality	虚拟现实
VRF	Virtual Routing Forwarding	虚拟路由转发

(续表)

缩略语	全 称	说 明
VRM	Virtual Resource Manager	虚拟资源管理器
vSGA	Virtual Shared Graphics Acceleration	虚拟共享图形加速
VSI	Virtual Subnet Identifier	虚拟子网标识符
VTEP	VxLAN Tunnel End Point	VxLAN隧道端点
vTPM	Virtualizing the Trusted Platform Module	虚拟化可信平台模块
VVOL	VMware Virtual Volume	VMware虚拟机卷
VxLAN	Virtual Extensible LAN	虚拟扩展局域网
WDDM	Windows Display Driver Model	桌面显示驱动模型
WI	Web Interface	网页界面
XPDM	Windows XP Display Driver Model	Windows XP显示驱动模型
YUV	Y”表示明亮度(Luminance、Luma), “U”和“V”则是色度、浓度(Chrominance、Chroma)	一种颜色编码方法

后记

本书在撰写过程中参考和引用了来自互联网和第三方公司或机构的内容，其中包括但不限于：美国国家标准与技术协会（NIST）关于云计算相关文件、Gartner相关报告、IDC相关报告、Nick McKeown等人于2008年在ACM SIGCOMM发表的题为OpenFlow: Enabling Innovation in Campus Networks的论文、Google公司发表的Data Center as a Computer一文、微软公司的WINDDK文档、中国移动云计算相关资料、埃森哲的《埃森哲2012年技术展望》白皮书、蒋清野对开源社区的跟踪研究（<http://www.qyjohn.net/?p=3399>）、Apache社区网站发布的文献资料（特别是关于CloudStack、OpenStack、Hadoop以及流处理相关的文档）、VMware公司网站公开文档、Cisco公司网站公开文档、H3C公司网站公开文档、Xen开源社区网站公开文档、Linux KVM社区网站公开文档、Intel公司网站公开文档、微软公司网站公开文档、OpenFlow网站文档、Oracle公司网站公开文档、VCE公司网站公开文档、Nutanix公司网站公开文档、HP公司网站公开文档、华为公司网站公开文档、Resource Allocation Algorithms for Virtualized Service Hosting Platforms（Mark Stillwell, 2010）、OpenDaylight网站公开文档、Wikipedia网站公开文档（特别是<http://zh.wikipedia.org/wiki/MapReduce>）、清华大学出版社的《分布式云数据中心的建设与管理》等。同时，我们在撰写过程中采用了互联网搜索工具（如Baidu和Google）搜索查阅了一些文章并引用了部分文字，已难以确定具体出处，无法一一列举。如读者发现未列明出处的引用，可通过出版社告知作者，在本书下一版中会补充到引用说明中或做其他修订处理。

在书中以及本书描述的产品中，出现的商标、产品名称、服务名称以及公司名称，由其各自的所有人拥有。本书内容不构成任何形式的承诺。除非适法要求，作者及出版社对本书所有内容不提供任何明示或暗示的保证。

在法律允许的范围内，本书作者及出版社在任何情况下都不对因使用本书相关内容而产生的任何特殊的、附带的、间接的、继发性的损害进行赔偿，也不对任何利润、数据、商誉或预期的损失进行赔偿。

附录

CDLink: <http://pan.baidu.com/s/1eQlljEa>

密码: **bm3d**

Table of Contents

[书名页](#)

[版权页](#)

[编委会](#)

[作者简介](#)

[内容简介](#)

[序言一](#)

[序言二](#)

[前言](#)

[目录](#)

[第1章 云计算理念的发展](#)

[1.1 云计算的基础概念与架构](#)

[1.2 云计算的发展趋势](#)

[第2章 云计算的架构内涵与关键技术](#)

[2.1 云计算的总体架构](#)

[2.1.1 云计算核心架构上下文](#)

[2.1.2 云计算平台架构](#)

[2.2 云计算架构的关键技术](#)

[2.2.1 超大规模资源调度算法](#)

[2.2.2 异构集成技术](#)

[2.2.3 应用无关的可靠性保障技术](#)

[2.2.4 单VM及多VM的弹性伸缩技术](#)

[2.2.5 计算近端I/O性能加速技术](#)

[2.2.6 网络虚拟化技术](#)

[2.2.7 应用管理自动化技术](#)

[2.3 云计算核心架构的竞争力衡量维度](#)

[2.3.1 低TCO](#)

[2.3.2 弹性伸缩](#)

[2.3.3 高性能](#)

[2.3.4 领先的用户体验](#)

[2.3.5 高安全](#)

[2.3.6 高可靠](#)

[2.4 云计算解决方案的典型架构组合及落地应用场景](#)

[2.4.1 桌面云](#)

[2.4.2 存储云](#)

[2.4.3 IDC托管云](#)

[2.4.4 企业私有云](#)

[2.4.5 大数据分析云](#)

[2.4.6 数据库云](#)

[2.4.7 媒体云](#)

[2.4.8 电信NFV云](#)

[第3章 云计算相关的开源软件](#)

[3.1 云计算领域开源软件概览](#)

[3.2 Cloud OS开源软件：CloudStack](#)

[3.2.1 CloudStack的总体架构](#)

[3.2.2 CloudStack的资源管理](#)

[3.2.3 CloudStack的虚拟机管理](#)

[3.2.4 CloudStack的块存储管理](#)

[3.2.5 CloudStack的虚拟网络](#)

[3.3 Cloud OS开源软件：OpenStack](#)

[3.3.1 OpenStack的总体架构](#)

[3.3.2 OpenStack的计算服务：Nova](#)

[3.3.3 OpenStack的块存储服务：Cinder](#)

[3.3.4 OpenStack的网络服务：Neutron](#)

[3.3.5 OpenStack的镜像服务：Glance](#)

[3.3.6 OpenStack的身份服务：KeyStone](#)

[3.4 开源和社区发展](#)

[3.4.1 Hypervisor社区发展](#)

[3.4.2 Cloud OS社区发展](#)

[3.5 开源还是闭源](#)

[第4章 面向电信及企业关键应用的计算虚拟化](#)

[4.1 计算虚拟化核心引擎：Hypervisor介绍](#)

[4.1.1 业界典型计算虚拟化架构说明](#)

[4.1.2 满足电信和企业关键应用的计算虚拟化技术](#)

[4.2 跨服务器的计算资源调度算法](#)

[4.2.1 高性能、低时延的虚拟机热迁移机制](#)

[4.2.2 计算资源池的动态资源调度管理和动态能耗管理](#)

[4.3 计算高可靠性保障](#)

[4.3.1 基于冷备机制的虚拟机HA保护](#)

[4.3.2 基于热备机制的虚拟机运行业务镜像冗余方案](#)

[4.3.3 无状态计算及物理机可靠性保障](#)

[第5章 面向网络自动化、多租户的网络虚拟化](#)

[5.1 网络虚拟化的驱动力与关键需求](#)

[5.2 SDN架构](#)

[5.2.1 IETF定义的SDN架构介绍](#)

[5.2.2 ONF OpenFlow网络架构](#)

[5.2.3 OpenFlow协议介绍](#)

[5.2.4 OF-Config](#)

[5.2.5 ONF及OpenDayLight标准联盟](#)

[5.3 网络虚拟化关键技术：大二层实现](#)

[5.3.1 CT流派：以交换机为中心的大二层技术](#)

[5.3.2 IT流派：以服务器叠加网为中心的Overlay技术](#)

[5.4 网络虚拟化关键技术：多租户网络实现](#)

[5.5 网络虚拟化端到端解决方案](#)

[5.6 网络云化还有多远](#)

[第6章 面向企业关键应用性能提升和存储管理简化的存储虚拟化](#)

[6.1 云计算的存储虚拟化概述](#)

[6.2 灵活的软件定义存储](#)

[6.3 传统存储SAN/NAS的虚拟化](#)

[6.4 分布式存储池化和加速](#)

[6.4.1 分布式存储概述](#)

[6.4.2 分布式存储系统的架构](#)

[6.4.3 分布式存储关键技术：性能提升技术](#)

[6.4.4 分布式存储关键技术：简化管理技术](#)

[6.4.5 分布式存储关键技术：安全可靠增强技术](#)

[第7章 云接入的关键技术架构与应用](#)

[7.1 云接入的概述](#)

[7.1.1 什么是云接入](#)

[7.1.2 云接入的作用和意义](#)

[7.1.3 云接入的挑战和需求](#)

[7.2 云接入的架构](#)

[7.3 云接入的典型应用](#)

[7.3.1 桌面云的概念和价值](#)

[7.3.2 桌面云的逻辑架构](#)

[7.3.3 桌面云典型应用场景](#)

[7.3.4 移动办公的概念和价值](#)

[7.3.5 移动办公的逻辑架构](#)

[7.3.6 移动办公解决方案的特点](#)

[7.4 云接入的关键技术](#)

[7.4.1 桌面云协议简介](#)

[7.4.2 桌面云协议关键技术：高效远程显示](#)

[7.4.3 桌面云协议关键技术：低资源消耗的多媒体视频](#)

[7.4.4 桌面云协议关键技术：低时延音频](#)

[7.4.5 桌面云协议关键技术：兼容多种外设](#)

[7.4.6 桌面云协议总结与其他实现](#)

[7.5 云接入的发展趋势](#)

[7.5.1 云接入的未来发展](#)

[7.5.2 VDI](#)

[7.5.3 DaaS](#)

[7.5.4 移动办公](#)

[第8章 云管理与自动化的关键技术架构与应用](#)

[8.1 业务应用驱动的拉通计算、存储、网络自动化](#)

[8.1.1 自动化部署](#)

[8.1.2 动态调度](#)

[8.1.3 网络自动化](#)

[8.2 物理和虚拟化资源的统一管控](#)

[8.2.1 物理资源管理](#)

[8.2.2 虚拟化资源管理](#)

[8.2.3 资源集群管理](#)

[8.2.4 虚拟机资源管理](#)

[8.3 基于网络的硬件即插即用的自动化机制](#)

[8.3.1 设备自动发现和部署](#)

[8.3.2 服务器自动化](#)

[8.4 异构硬件的统一接入管理](#)

[8.5 服务目录和应用管理](#)

[8.5.1 应用发布流程介绍](#)

[8.5.2 应用管理原理](#)

[8.6 面向云管理的ITSM](#)

[8.7 云平台第三方App资源使用计量](#)

[8.8 云管理的应用案例](#)

[8.8.1 M运营商私有云建设](#)

[8.8.2 T运营商分布式数据中心](#)

[8.8.3 新加坡S运营商中小企业IT应用托管](#)

[第9章 云安全架构与应用实践](#)

[9.1 端到端云安全架构](#)

[9.1.1 云计算中的主要安全威胁](#)

[9.1.2 端对端的安全架构](#)

[9.2 可信计算TPM/vTPM](#)

[9.2.1 TPM功能1：主机启动/静态度量](#)

[9.2.2 TPM功能2：虚拟机的静态度量](#)

[9.2.3 TPM功能3：主机动态度量](#)

[9.2.4 TPM功能4：VM动态度量](#)

[9.2.5 TPM功能5：远程证明](#)

[9.3 虚拟机的安全隔离](#)

[9.3.1 vCPU调度隔离安全](#)

[9.3.2 内存隔离](#)

[9.3.3 内部网络隔离](#)

[9.3.4 磁盘I/O隔离](#)

[9.3.5 用户数据隔离](#)

[9.4 虚拟化环境中的网络安全](#)

[9.4.1 虚拟交换机及防ARP攻击](#)

[9.4.2 IP/MAC防欺骗功能设计](#)

[9.4.3 VLAN](#)

[9.5 云数据安全](#)

[9.5.1 云存储加密与用户数据安全](#)

[9.5.2 用户数据安全有效保护](#)

[9.6 公有云、私有云的安全组](#)

[9.7 云安全管理](#)

[9.7.1 日志管理](#)

[9.7.2 账户和密码安全](#)

[9.7.3 分权分域管理](#)

[9.8 云安全应用实施案例](#)

[9.9 云计算安全的其他考虑](#)

[第10章 大数据平台核心技术与架构](#)

[10.1 大数据特点与支撑技术](#)

[10.1.1 数据采集技术](#)

[10.1.2 数据预处理技术](#)

[10.1.3 数据存储及管理技术](#)

[10.1.4 数据分析及挖掘技术](#)

[10.1.5 数据展现与应用技术](#)

[10.2 企业级Hadoop](#)

[10.2.1 Apache Hadoop起源](#)

[10.2.2 企业级Hadoop总体框架](#)

[10.2.3 HDFS](#)

[10.2.4 MapReduce](#)

[10.2.5 ZooKeeper](#)

[10.2.6 HBase](#)

[10.2.7 Hive](#)

[10.3 流处理技术](#)

[10.3.1 流处理的应用场景](#)

[10.3.2 流处理技术的关键概念](#)

[10.3.3 流处理技术的辨析](#)

[10.3.4 流处理技术的最新发展](#)

[10.3.5 分布式事件的流处理技术](#)

[10.4 大数据在金融领域的探索与实践](#)

[10.4.1 银行业现状和大数据的潜在机会](#)

[10.4.2 大数据时代的银行业发展](#)

[10.4.3 大数据在银行业的发展趋势](#)

[10.4.4 大数据在金融行业的实践](#)

[10.5 未来大数据应用畅想](#)

[10.5.1 身边的大数据](#)

[10.5.2 大数据将重构很多行业的商业思维和商业模式](#)

[第11章 企业私有云和公有云对IAAS层的诉求](#)

[11.1 企业私有云和公有云对IAAS层的诉求](#)

[11.2 一体机的市场和技术](#)

[11.2.1 一体机市场](#)

[11.2.2 一体机技术](#)

[11.2.3 一体机产品介绍](#)

[11.3 一体机市场、技术趋势](#)

[11.3.1 一体机市场趋势](#)

[11.3.2 一体机技术趋势](#)

[结语](#)

[缩略语](#)

[后记](#)

[附录CD](#)